

BAB V

IMPLEMENTASI SISTEM

Pada bab V akan menjelaskan tentang desain program dan coding program. Berikut ini tampilan-tampilan halaman yang ada didalam program dibagi menjadi 2 bagian yaitu user dan admin.

5.1. Spesifikasi Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi adalah:

1. AMD Dual-Core A9-9420 APU (3.00 – 3.60 GHz, 1 MB Cache)
2. Memory 4.00 GB
3. Hardisk 1 TB
4. Monitor 14”

5.2. Spesifikasi Perangkat Lunak

1. Android Studio
2. Sublime Text 3
3. SQLyog
4. XAMPP v3.2.1
5. Google Chrome

5.3. Batasan Implementasi Sistem

Tahapan implementasi pada sistem ini merupakan kelanjutan dari tahapan perancangan sistem yang telah di uraikan sebelumnya pada bab IV, pada bab ini juga akan diuraikan implementasi proses sistem dari perancangan sistem yang telah dibuat sebelumnya.

5.4. Implementasi Sistem

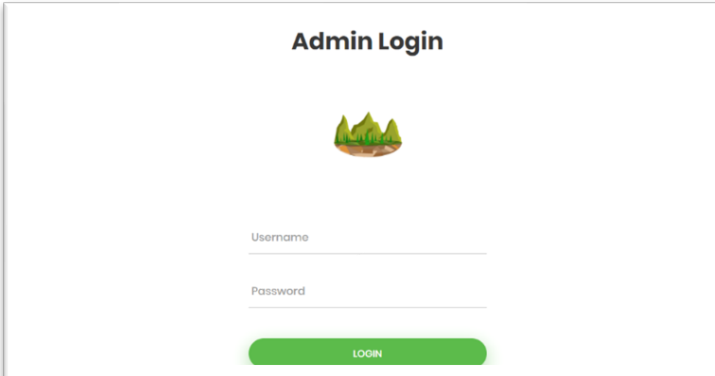
Implementasi pada sistem ini memiliki beberapa proses yang sudah di uraikan pada bab VI terbagi menjadi beberapa proses dan form, sourcode dari implementasi sistem ini lampirkan pada bab ini.

5.5. Implementasi Antarmuka

Implementas antarmuka merupakan tampilan sistem secara keseluruhan dan jika sistem pertama kali dijalankan maka akan tampil adalah form login pada admin seperti pada gambar 5.5.1 di bawah ini:

5.5.1. Halaman Login Admin

Halaman login merupakan halaman hak akses admin untuk masuk kedalam halaman utama, Didalam halaman login dibutuhkan username dan password untuk memverifikasikan kebenaran kepemilikan hak akses. Untuk lebih jelasnya dapat dilihat pada gambar 5.1



The image shows a web form titled "Admin Login". At the top center, the text "Admin Login" is displayed in a bold, black font. Below this title is a small, stylized logo of a green tree with a brown base. Underneath the logo, there are two input fields: the first is labeled "Username" and the second is labeled "Password". Both fields have a light gray border and a white background. At the bottom center of the form, there is a prominent green button with the word "LOGIN" written in white capital letters.

Gambar 5.1. Halaman Login

5.5.1.1. Coding Program Halaman Login Admin

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

    public function index()
    {
        //$data['content'] = 'content/content';
        // $this->load->view('template/content', $data);
        $this->load->view('login/index');
    }
}

```

a. Auth Halaman Login dan Logout

```

<?php
class auth extends CI_controller {

    function __construct(){
        parent::__construct();
    }
    function login (){
        // proses login disini
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $jb = "";
        $us = "";
        //$hasil = $this->m_login->login($username,$password)->result();
        $hasil = $this->db->query("SELECT * FROM tbl_admin
WHERE username = '$username' AND password = '$password'")->result();
        foreach ($hasil as $k) {
            $id = $k->id_admin;
            $nm = $k->nama_admin;
            $jb = $k->akses;
        }
        if(empty($id)){
            $this->session->sess_destroy();
            redirect('Welcome');
        }

        $this->session->set_userdata(
            array(
                'id'=>$id,
                'jb'=>$jb,
                'nm'=>$nm,
                'status_login'=>'oke'
            )
        )
    }
}

```

```

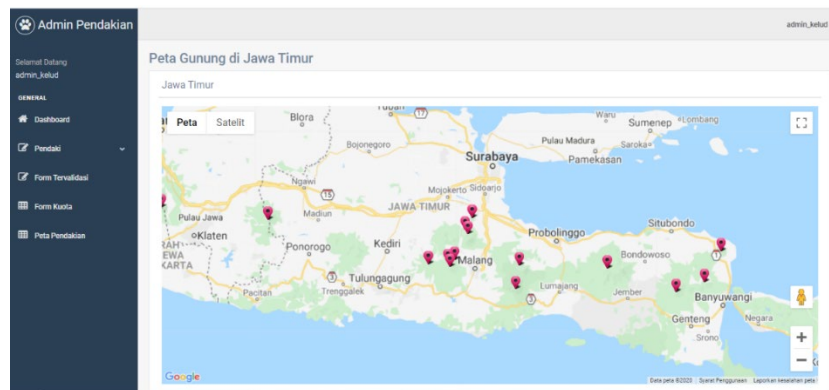
    ));
    $this->session->set_userdata('notif','1');
    redirect('dashboard'); }

function logout (){
    $this->session->sess_destroy();
    redirect('auth/login');}} ?>

```

5.5.2. Halaman Dashboard Admin

Halaman dashboard admin merupakan halaman yang pertama kali akan muncul jika sudah masuk dari halaman login. Halaman dashboard hanya memunculkan suatu maps beserta koordinat gunung yang masuk ke sistem booking. Berikut ini desain halaman dashboard admin dari sistem ini.



Gambar 5.2. Halaman Dashboard Admin

5.5.2.1. Coding Program Halaman Dashboard Admin

a. View Halaman Dashboard Admin

```

<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA2JUy7LvRp3vwLVJ
xf0TIYyRqSSTpo3kU&callback=initMap">
</script>
<script>
function initMap() {
    var uluru = {lat: -7.8039901, lng: 112.7380089};
    var map = new google.maps.Map(document.getElementById('map'), {
        zoom: 8.5,
        center: uluru
    });
}

```

```

//utk membuat data marker
var features = [
  <?php foreach($gunung->result() as $row){ ?>
    {
      position: new google.maps.LatLng(<?php echo $row->latitude;?>,<?php
echo $row->longitude;?>),
      type: 'info',
      title : '<?php echo $row->nama_gunung;?>'
    },
  <?php } ?>
];
features.forEach(function(feature) {
  <?php foreach ($gunung->result() as $row) { ?>
    var marker = new google.maps.Marker({
      position: new google.maps.LatLng(<?php echo $row->latitude;?>,<?php
echo $row->longitude;?>),
      icon: "assets/merah.png",
      title: '<?php echo $row->nama_gunung;?>',
      map: map
    });
    <?php
  } ?>
});
}
google.maps.event.addDomListener(window, 'load', init);
</script>
<div class="right_col" role="main">
  <div class="">
    <div class="page-title">
      <div class="title_left">
        <h3>Peta Gunung di Jawa Timur</h3>
      </div>
    </div>
    <div class="clearfix"></div>
    <div class="row">
      <div class="col-md-12 col-sm-12 col-xs-12">
        <div class="x_panel">
          <div class="x_title">
            <h2>Jawa Timur</h2>
            <div class="clearfix"></div>
          </div>
          <div class="x_content">
            <div id="map" style="width:100%;height:500px"></div>
          </div> </div> </div> </div> </div> </div>

```

b. Controller Halaman Dashboard Admin

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Dashboard extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('M_data');
    }

    public function index(){
        $data['content'] = 'content/dashboard';
        $data['title']    = 'Dashboard';
        $data['gunung'] = $this->M_data->get_data('tbl_gunung');

        $this->load->view('template/content', $data);
    }
}

```

5.5.3. Halaman Form Validasi Admin

Halaman form validasi admin merupakan halaman yang digunakan untuk menampung semua data user yang mendaftar booking online. Jika data user benar maka akan divalidasi oleh admin dan sebaliknya jika ada user belum lengkap maka data tidak akan divalidasi. Berikut ini desain halaman form validasi admin dari sistem ini.

No	Tanggal Berangkat	Tanggal Pulang	ID KTP	Nama Pendaki	Tanggal Lahir	Alamat	Kebangsaan	No HP	Pekerjaan
1	2020-08-17	2020-08-18	12345678901	Cici	1996-08-20	Sambutan	WNI	085321698452	Pegawai
2	2020-08-20	2020-08-21	12345678906	Deni	1996-01-24	Rembang	WNI	085412369852	Pelajar
3	2020-08-24	2020-08-25	1234567854	Lulu	1999-01-21	Bandung	WNI	085123698523	Pelajar
4	2020-08-20	2020-08-21	23452345345	Ruru	1995-05-27	Surabaya	WNI	08554236985	Pegawai
5	2020-09-01	2020-09-02	3515132207900010	Fandi aja testing	1996-06-20	sambutanrhg	wni	083856829002	mahasiswa
6	2020-08-18	2020-08-20	12345678905	Laila	1996-08-25	Surabaya	WNI	085136498523	Pelajar
7	2020-08-20	2020-08-22	12345678908	Susy	1996-05-02	Sidoarjo	WNA	085236985236	Wirasaha
8	2020-08-26	2020-08-28	456456456	Yuli	1995-05-21	Twati	WNI	085123698563	Pegawai
9	2020-08-17	2020-08-19	12345678902	Rere	1996-08-21	Sidoarjo	WNI	081236542896	Mahasiswa
10	2020-08-27	2020-08-29	1234567898	Jeno	1995-05-15	Maleno	WNI	085147369236	Pegawai

Gambar 5.3. Halaman Form Validasi Admin

5.5.3.1. Coding Program Halaman Form Validasi Admin

a. View Form Validasi Admin

```

<div class="x_content">
    <table id="datatable-responsive" class="table table-striped table-
bordered dt-responsive nowrap" cellspacing="0" width="100%">
        <thead>
            <tr>
                <th>No</th>
                <th>Tanggal <br>Berangkat</th>
                <th>Tanggal <br>Pulang</th>
                <th>ID<br>KTP </th>
                <th>Nama<br>Pendaki</th>
                <th>Tanggal<br>Lahir</th>
                <th>Alamat</th>
                <th>Kebangsaan</th>
                <th>No HP</th>
                <th>Pekerjaan</th>
                <th>Pembayaran</th>
                <th>Hari Mendaki</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <?php
                $jb = $this->session->userdata('jb');
                $no = 1;
                $hari = array();
                foreach ($pengunjung->result() as $row) {
                    $date1 = date("Y-m-d", strtotime($row->tgl_berangkat));

```

```

$date2 = date("Y-m-d", strtotime($row->tgl_pulang));
$start_date = new DateTime($date1);
$end_date = new DateTime($date2);
$interval = $start_date->diff($end_date);
$jml = count($hari);
for ($a=0;$a<$jml;$a++){
    for ($b=$a;$b<$jml;$b++){
        if ($hari[$b] < $hari[$a]){
            $tmp = $hari[$b];
            $tmp1 = $idp[$b];
            $hari[$b] = $hari[$a];
            $idp[$b] = $idp[$a];
            $hari[$a] = $tmp;
            $idp[$a] = $tmp1;
        } } }
for ($c=0; $c < $jml ; $c++) {
    $dt = $this->db->query("
SELECT * FROM tbl_pengunjung tp
JOIN tbl_booking tb ON tb.kode_booking = tp.kode_booking
JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung
WHERE
status_validasi = '0'
AND tp.id_pengunjung = '$idp[$c]'
AND tg.nama_gunung = '$jb'
")->result();
foreach ($dt as $drow) {
?>
<tr>

```



```

<td><?php echo $no++; ?></td>
<td><?php echo $drow->tgl_berangkat; ?></td>
<td><?php echo $drow->tgl_pulang; ?></td>
<td><?php echo $drow->id_ktp; ?></td>
<td><?php echo $drow->nama_pengunjung; ?></td>
<td><?php echo $drow->tanggal_lahir; ?></td>
<td><?php echo $drow->alamat; ?></td>
<td><?php echo $drow->kebangsaan; ?></td>
<td><?php echo $drow->no_hp; ?></td>
<td><?php echo $drow->pekerjaan; ?></td>
<td><?php echo $drow->status_pembayaran; ?></td>
<td><?php echo $hari[$c];?> Hari</td>
<td>
    <div class="btn-group" align="center">
        <a class="btn btn-success" href="<?php echo
base_url("Tampilan_validasi/btn_validasi/".$drow->id_pengunjung); ?>"><i
class="fa fa-check"></i></a>
        <a class="btn btn-danger" href="<?php echo
base_url("Tampilan_validasi/btn_hapus/".$drow->id_pengunjung); ?>"
onclick="return confirm("Yakin ingin menghapus data ini?")"><i class="fa fa-
trash"></i></a>
    </div>
</td>
</tr>
<?php }}
?>
</tbody>
</table>
</tbody>
</table> </div> </div> </div> </div> </div>

```

b. Controller Form Validasi Admin

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
class Tampilan_validasi extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('M_data');
        require APPPATH.'libraries/phpmailer/src/Exception.php';
require APPPATH.'libraries/phpmailer/src/PHPMailer.php';
require APPPATH.'libraries/phpmailer/src/SMTP.php';
    }

    public function index(){
        $jb = $this->session->userdata('jb');
        $data['content'] = 'content/tampilan_validasi';
        $data['title'] = 'Form Validasi';
        //$data['pengunjung'] =
$this->M_data->get_data_where('tbl_pengunjung', array('status_validasi' => '0'));
        $data['pengunjung'] = $this->db->query("
SELECT *
FROM
tbl_pengunjung tp
JOIN tbl_booking tb ON tb.kode_booking = tp.kode_booking
JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung
WHERE
tg.nama_gunung = '$jb'
AND status_validasi = '0'");

        $this->load->view('template/content', $data);
    }

    public function btn_validasi($id)
    {
        $table = 'tbl_pengunjung';
        $data = array('status_validasi' => '1');
        $where = array('id_pengunjung' => $id);

        $this->M_data->update_data($table,$data,$where);
        $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking WHERE tp.id_pengunjung =
'$id'")->result();

```

```

        foreach ($data as $dt) {
            $email = $dt->email;
            $nm = $dt->nama_pengunjung;
            $by = number_format($dt->biaya);
            $idb = $dt->kode_booking;
        }

    // PHPMailer object
    $response = false;
    $mail = new PHPMailer();

    // SMTP configuration
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'muhammadfarid031@gmail.com';
    $mail->Password = 'mARKamAH123';
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    $mail->setFrom('muhammadfarid031@gmail.com', 'Administrasi Pendaki Jawa
Timur');
    $mail->addReplyTo('muhammadfarid031@gmail.com', 'Administrasi Pendaki
Jawa Timur');

    // Add a recipient
    $mail->addAddress($email); //email tujuan pengiriman email

    // Email subject
    $mail->Subject = 'Informasi Booking Pendaki';

    // Set email format to HTML
    $mail->isHTML(true);

    // Email body content
    $mailContent = "<h1>Data Booking Anda Telah Lolos Validasi</h1>
        <p>Hai ".$nm." terima kasih telah menunggu, Data anda telah lolos
validasi. Silahkan melakukan pembayaran sebesar Rp. ".$by.". Setelah melakukan
pembayaran harap melakukan verifikasi pembayaran melalui aplikasi mobile dengan
memasukkan kode booking anda. Kode booking anda adalah ".$idb."</p>";
    $mail->Body = $mailContent;

    // Send email
    if(!$mail->send()){
        //echo 'Message could not be sent.';
    }

```

```

        echo 'Mailer Error: ' . $mail->ErrorInfo;
    }else{
        //echo 'Message has been sent';
    }

    redirect('Tampilan_validasi'); }

    public function btn_hapus($id)
    {
        $table = 'tbl_pengunjung';
        $where = array('id_pengunjung' => $id);

        $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking WHERE tp.id_pengunjung =
'$id'")->result();
        foreach ($data as $dt) {
            $email = $dt->email;
            $nm = $dt->nama_pengunjung;
            $by = number_format($dt->biaya);
            $idb = $dt->kode_booking;
        }

        // PHPMailer object
        $response = false;
        $mail = new PHPMailer();
        // SMTP configuration
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com'; //sesuaikan sesuai nama domain
hosting/server yang digunakan
        $mail->SMTPAuth = true;
        $mail->Username = 'pendaftaranpendaki@gmail.com'; // user email
        $mail->Password = 'pendaki123'; // password email
        $mail->SMTPSecure = 'ssl';
        $mail->Port = 465;

        $mail->setFrom('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki Jawa
Timur');
        $mail->addReplyTo('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki
Jawa Timur');

        // Add a recipient
        $mail->addAddress($email); //email tujuan pengiriman email

        // Email subject
        $mail->Subject = 'Informasi Booking Pendaki';
    }
}

```

```

// Set email format to HTML
$mail->isHTML(true);

// Email body content
$mailContent = "<h1>Data Booking Anda Tidak Lolos Validasi</h1>
                <p>Hai ".$nm." terima kasih telah menunggu, Mohon maaf data anda
                tidak lolos validasi. Silahkan melakukan booking ulang dengan data yang valid.</p>";
$mail->Body = $mailContent;

// Send email
if(!$mail->send()){
    //echo 'Message could not be sent.';
    //echo 'Mailer Error: ' . $mail->ErrorInfo;
}else{
    //echo 'Message has been sent';
}
$table2 = 'tbl_booking';
$where2 = array('kode_booking' => $idb);
$this->M_data->hapus_data($table,$where);
$this->M_data->hapus_data($table2,$where2);
        redirect('Tampilan_validasi');
    }
}

```

5.5.4. Halaman Form Konfirmasi Pembayaran Admin

Halaman form konfirmasi pembayaran admin merupakan halaman yang digunakan untuk menampung semua data user yang membayar nominal yang sudah ditentukan oleh user. Jika sudah masuk ke sistem, langsung dikonfirmasi oleh admin. Berikut ini desain halaman form konfirmasi pembayaran admin dari sistem ini.

No	ID Konfirmasi	Kode Booking	Tanggal Transfer	Atas Nama	Nama Bank	Nominal Transfer	No Rekening	Status Konfirmasi
1	8	379	2020-08-20	Yuli	BCA	Rp. 10.000	2147483647	Belum Konfirmasi
2	9	396	2020-08-20	yuyu	bni	Rp. 10.000	1263673788	Belum Konfirmasi

Gambar 5.4. Halaman Konfirmasi Pembayaran Admin

5.5.4.1. Coding Program Halaman Form Konfirmasi Pembayaran Admin

a. View Form Konfirmasi Pembayaran Admin

```

<div class="x_content">
  <table id="datatable-responsive" class="table table-striped table-bordered
dt-responsive nowrap" cellspacing="0" width="100%">
    <thead>
      <tr>
        <th>No</th>
        <th>ID Konfirmasi </th>
        <th>Kode Booking</th>
        <th>Tanggal Transfer</th>
        <th>Atas Nama</th>
        <th>Nama Bank</th>
        <th>Nominal Transfer</th>
        <th>No Rekening</th>
        <th>Status Konfirmasi</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
      <?php
        $no = 1;
        foreach ($konfirmasi->result() as $row) { ?>
          <tr>
            <td><?php echo $no++; ?></td>
            <td><?php echo $row->id_konfirmasi; ?></td>
            <td><?php echo $row->kode_booking; ?></td>

```

```

<td><?php echo $row->tgl_tf; ?></td>
<td><?php echo $row->nama_tf; ?></td>
<td><?php echo $row->bank_tf; ?></td>
<td>Rp. <?php echo number_format($row->biaya_tf); ?></td>
<td><?php echo $row->no_rek_tf; ?></td>
<td><?php if ($row->status_konfirmasi == '1'){echo
"Konfirmasi"; } else {echo "Belum Konfirmasi"; } ?></td>
<td>
<div class="btn-group" align="center">
<a class="btn btn-success" href="<?php echo
base_url('Tampilan_form_pembayaran/btn_validasi/'.$row->id_konfirmasi); ?>"><i
class="fa fa-check"></i></a>
<a class="btn btn-danger" href="<?php echo
base_url('Tampilan_form_pembayaran/btn_hapus/'.$row->id_konfirmasi); ?>"
onclick="return confirm('Yakin ingin menghapus data ini?')"><i class="fa fa-
trash"></i></a>
</div>
</td>
</tr>
<?php }
?>
</tbody>

```

b. Controller Form Konfirmasi Pembayaran Admin

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
class Tampilan_form_pembayaran extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('M_data');
        require APPPATH.'libraries/phpmailer/src/Exception.php';
        require APPPATH.'libraries/phpmailer/src/PHPMailer.php';
        require APPPATH.'libraries/phpmailer/src/SMTP.php';
    }

    public function index(){
        $jb = $this->session->userdata('jb');
        $data['content'] = 'content/tampilan_form_pembayaran';
        $data['title'] = 'Form Pembayaran';
        // $data['konfirmasi'] = $this->M_data->get_data('tbl_konfirmasi');
    }
}

```

```

        $data['konfirmasi'] = $this->db->query("SELECT * FROM
tbl_konfirmasi tk JOIN tbl_booking tb ON tb.kode_booking = tk.kode_booking JOIN
tbl_gunung tg ON tg.id_gunung = tb.id_gunung WHERE tg.nama_gunung = '$jb'
AND tk.status_konfirmasi = '0'");

        $this->load->view('template/content', $data);
    }
    public function btn_validasi($id)
    {
        $table = 'tbl_konfirmasi';
        $data = array('status_konfirmasi' => '1');
        $where = array('id_konfirmasi' => $id);
        $this->M_data->update_data($table,$data,$where);
        $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking JOIN tbl_konfirmasi tk ON
tk.kode_booking = tb.kode_booking WHERE tk.id_konfirmasi = '$id'")->result();
        foreach ($data as $dt) {
            $email = $dt->email;
            $nm = $dt->nama_pengunjung;
            $by = number_format($dt->biaya);
            $idb = $dt->kode_booking; }

        // PHPMailer object
        $response = false;
        $mail = new PHPMailer();

        // SMTP configuration
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'muhammadfarid031@gmail.com';
        $mail->Password = 'mARKamAH123';
        $mail->SMTPSecure = 'tls';
        $mail->Port = 587;

        $mail->setFrom('muhammadfarid031@gmail.com', 'Administrasi Pendaki Jawa
Timur');
        $mail->addReplyTo('muhammadfarid031@gmail.com', 'Administrasi Pendaki
Jawa Timur');

        // Add a recipient
        $mail->addAddress($email); //email tujuan pengiriman email

        // Email subject
        $mail->Subject = 'Informasi Booking Pendaki';

```



```

// Set email format to HTML
$mail->isHTML(true);

// Email body content
$mailContent = "<h1>Data Booking Anda Telah Terbayar</h1>
                <p>Hai ".$nm." terima kasih telah mengikuti prosedur, Data anda telah
lolos validasi dan telah terbayar. Silahkan menunjukkan email ini sebagai bukti bahwa
anda telah mendaftar secara online. Selamat Mendaki, Keep safety and be fun.</p>";
$mail->Body = $mailContent;

// Send email
if(!$mail->send()){
    //echo 'Message could not be sent.';
    //echo 'Mailer Error: ' . $mail->ErrorInfo;
}else{
    //echo 'Message has been sent';
}
$stable1 = 'tbl_booking';
$data1 = array('status_lunas' => '1');
$where1 = array('kode_booking' => $idb);

$this->M_data->update_data($stable1,$data1,$where1);

$stable2 = 'tbl_pengunjung';
$data2 = array('status_pembayaran' => '1');
$where2 = array('kode_booking' => $idb);

$this->M_data->update_data($stable2,$data2,$where2);
redirect('Tampilan_form_pembayaran');
}
public function btn_hapus($id)
{
    $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking JOIN tbl_konfirmasi tk ON
tk.kode_booking = tb.kode_booking WHERE tk.id_konfirmasi = '$id'")->result();
    foreach ($data as $dt) {
        $email = $dt->email;
        $nm = $dt->nama_pengunjung;
        $by = number_format($dt->biaya);
        $idb = $dt->kode_booking; }

// PHPMailer object
$response = false;
$mail = new PHPMailer();

```

```

// SMTP configuration
$mail->isSMTP();
$mail->Host    = 'smtp.gmail.com'; //sesuaikan sesuai nama domain
hosting/server yang digunakan
$mail->SMTPAuth = true;
$mail->Username = 'pendaftaranpendaki@gmail.com'; // user email
$mail->Password = 'pendaki123'; // password email

$mail->SMTPSecure = 'ssl';
$mail->Port    = 465;
$mail->setFrom('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki Jawa
Timur');
$mail->addReplyTo('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki
Jawa Timur');

// Add a recipient
$mail->addAddress($email); //email tujuan pengiriman email
// Email subject
$mail->Subject = 'Informasi Booking Pendaki';
// Set email format to HTML
$mail->isHTML(true);
// Email body content
$mailContent = "<h1>Data Booking Anda Tidak Terbayara</h1>
                <p>Hai ".$nm." terima kasih telah mengikuti prosedur, Data booking tidak
terbayar. Silahkan melakukan booking ulang dengan data yang valid.</p>";
$mail->Body = $mailContent;
// Send email
if(!$mail->send()){
    //echo 'Message could not be sent.';
    //echo 'Mailer Error: ' . $mail->ErrorInfo;
}else{
    //echo 'Message has been sent';
}
$table = 'tbl_konfirmasi';
$where = array('id_konfirmasi' => $id);
$table1 = 'tbl_booking';
    $where1 = array('kode_booking' => $idb);

    $table2 = 'tbl_pengunjung';
    $where2 = array('kode_booking' => $idb);

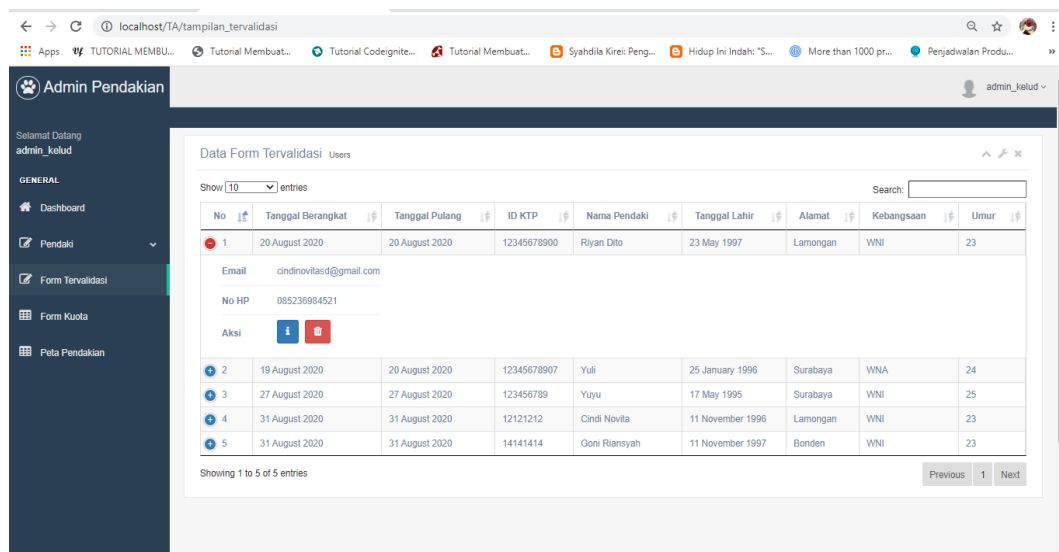
$this->M_data->hapus_data($table,$where);
$this->M_data->hapus_data($table2,$where2);

```

```
$this->M_data->hapus_data($table1,$where1);
redirect('Tampilan_form_pembayaran'); } }
```

5.5.5. Halaman Form Tervalidasi Admin

Halaman form tervalidasi admin merupakan halaman yang digunakan untuk menampung semua data user yang sudah valid. Berikut ini desain halaman form tervalidasi admin dari sistem ini.



Gambar 5.5. Halaman Form Tervalidasi Admin

5.5.5.1. Coding Program Halaman Form Tervalidasi Admin

a. View Form Tervalidasi Admin

```
<div class="x_content">
  <table id="datatable-responsive" class="table table-striped table-bordered
dt-responsive nowrap" cellspacing="0" width="100%">
  <thead>
  <tr>
  <th>No</th>
  <th>Tanggal Berangkat</th>
  <th>Tanggal Pulang</th>
  <th>ID KTP </th>
  <!-- <th>Leader</th>
  <th>ID KTP Leader</th> -->
  <th>Nama Pendaki</th>
  <th>Tanggal Lahir</th>
```

```

<th>Alamat</th>
<th>Kebangsaan</th>
<th>Umur</th>
<th>Email</th>
<th>No HP</th>
<th>Aksi</th>
<!-- <th>No HP Orang Tua</th>
<th>Pekerjaan</th>
<th>Porter</th>
<th>Pembayaran</th> -->
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach ($pengunjung->result() as $row) { ?>
<tr>
<td><?php echo $no++; ?></td>
<td><?php echo date("d F Y",
strtotime($row->tgl_berangkat)); ?></td>
<td><?php echo date("d F Y",
strtotime($row->tgl_pulang)); ?></td>
<td><?php echo $row->id_ktp; ?></td>
<!-- <td><?php if ($row->status_leader == '1'){echo "Leader"; }
else {echo "Anggota"; } ?></td>
<td><?php echo $row->id_ktp_leader; ?></td> -->
<td><?php echo $row->nama_pengunjung; ?></td>
<td><?php echo date("d F Y",
strtotime($row->tanggal_lahir)); ?></td>
<td><?php echo $row->alamat; ?></td>
<td><?php echo $row->kebangsaan; ?></td>
<td><?php echo $row->umur; ?></td>
<td><?php echo $row->email; ?></td>
<td><?php echo $row->no_hp; ?></td>
<td><button type="button" class="btn btn-primary" data-
toggle="modal" data-target="#myModal">Detail</button>
<a href="<?php echo
base_url('Tampilan_tervalidasi/hapus_data/'.$row->id_pengunjung); ?>"><button
class="btn btn-danger" title="Hapus Data" type="button" onclick="return
confirm('apakah Anda yakin ingin menghapus data ini?');"><i class="fa fa-
trash"></i></button></a> </td>

```

b. Controller Halaman Form Tervalidasi Admin

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Tampilan_tervalidasi extends CI_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('M_data');
    }
    public function index(){
        $jb = $this->session->userdata('jb');
        $data['content'] = 'content/tampilan_tervalidasi';
        $data['title'] = 'Form Tervalidasi';
        //$data['pengunjung'] =
        $this->M_data->get_data_where('tbl_pengunjung', array('status_pembayaran' => '1'));
        $data['pengunjung'] = $this->db->query("SELECT * FROM
tbl_pengunjung tp JOIN tbl_booking tb ON tb.kode_booking = tp.kode_booking JOIN
tbl_gunung tg ON tg.id_gunung = tb.id_gunung WHERE tg.nama_gunung = '$jb'
AND tp.status_pembayaran = '1'");

        $this->load->view('template/content', $data);
    }
    function hapus_data($id)
    {
        $table = 'tbl_pengunjung';
        $where = array('id_pengunjung' => $id);

        $sql = $this->db->query("SELECT * FROM tbl_pengunjung WHERE
id_pengunjung = '$id'")->result();
        foreach ($sql as $s) {
            $idb = $s->kode_booking;
        }
        $table1 = 'tbl_konfirmasi';
        $where1 = array('kode_booking' => $idb);
        $table2 = 'tbl_booking';
        $where2 = array('kode_booking' => $idb);

        $this->M_data->hapus_data($table1, $where1);
        $this->M_data->hapus_data($table2, $where2);
        $this->M_data->hapus_data($table, $where);

        redirect('Tampilan_tervalidasi');
    }
}

```

5.5.6. Halaman Form Kuota Admin

Halaman form kuota admin merupakan halaman yang digunakan untuk mengatur jumlah kuota dalam waktu yang ditentukan oleh admin. Berikut ini desain halaman form kuota admin dari sistem ini.

No	Nama Gunung	Tanggal Kuota	Jumlah Kuota	Cuaca	Status Kuota	Aksi
1	Kelud	2020-08-31	121	Sebagian Berawan	Buka	
2	Kelud	2020-09-01	98	Sebagian Berawan	Buka	
3	Kelud	2020-09-02	100	Sebagian Berawan	Buka	
4	Kelud	2020-09-03	100	Hujan	Buka	
5	Kelud	2020-09-04	100	Badai Petir	Buka	
6	Kelud	2020-09-05	100	Badai Petir Petang	Buka	
7	Kelud	2020-09-06	100	Sebagian Berawan	Buka	
8	Kelud	2020-09-07	100	Sebagian Berawan	Buka	

Gambar 5.6. Halaman Form Kuota Admin

5.5.6.1. Coding Program Halaman Form Kuota Admin

a. View Form Kuota Admin

```

<div class="x_content">
  <button type="button" class="btn btn-primary" data-toggle="modal" data-
target=".tambah_data">Tambah Data</button>
  <table id="datatable-responsive" class="table table-striped table-bordered
dt-responsive nowrap" cellspacing="0" width="100%">
    <thead>
      <tr>
        <th>No</th>
        <th>ID Kuota </th>
        <th>Nama Gunung</th>
        <th>Tanggal Kuota</th>
        <th>Jumlah Kuota</th>
        <th>Status Kuota</th>
        <th>Aksi</th>
      </tr>
    </thead>
    <tbody>

```

```

<?php
    $no = 1;
    foreach ($kuota as $row) { ?>
        <tr>
            <td><?php echo $no; ?></td>
            <td><?php echo $row->id_kuota; ?></td>
            <td><?php echo $row->nama_gunung; ?></td>
            <td><?php echo $row->tgl_kuota; ?></td>
            <td><?php echo $row->jumlah_kuota; ?></td>
            <td><?php if ($row->status_kuota == '1'){echo "Buka"; } else
{echo "Tutup"; } ?></td>
            <td>
                <a href="<?php echo
base_url('Tampilan_kuota/edit_data_page/'.$row->id_kuota); ?>"><button class="btn
btn-warning" title="Edit Data" data-toggle="modal" data-target=".edit_data"<?php
echo $no++; ?><i class="fa fa-edit"></i></button></a>
                <a href="<?php echo
base_url('Tampilan_kuota/hapus_data/'.$row->id_kuota); ?>"><button class="btn btn-
danger" title="Hapus Data" type="button" onclick="return confirm('apakah Anda
yakini ingin menghapus data ini?');"><i class="fa fa-trash"></i></button></a>
            </td>
        </tr>
    <?php }
    ?>
</tbody>
</table>
</tbody>
</table>
</div>

```

b. Controller Halaman Form Kuota Admin

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Tampilan_kuota extends CI_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('M_data');
    }
    public function index(){
        $jb = $this->session->userdata('jb');
        $data['content'] = 'content/tampilan_kuota';
        $data['title'] = 'Form Kuota';
    }
}

```

```

        //$data['kuota'] = $this->M_data->get_data('tbl_kuota');
        $data['gunung'] = $this->M_data->get_data('tbl_gunung');
        $data['kuota'] = $this->db->query("SELECT * FROM tbl_kuota tk
JOIN tbl_gunung tg ON tg.id_gunung = tk.id_gunung WHERE tg.nama_gunung =
'$jb'")->result();
        $this->load->view('template/content', $data);
    }
    function add_data()
    {
        $jb = $this->session->userdata('jb');
        $sql = $this->db->query("SELECT * FROM tbl_gunung WHERE
nama_gunung = '$jb'")->result();
        foreach ($sql as $s) {
            $idg = $s->id_gunung;
        }
        $args = $this->input->post();
        $table = 'tbl_kuota';
        $data = array(
            'id_gunung'           => $idg,
            'tgl_kuota'           => $args['tanggal_kuota'],
            'jumlah_kuota'       => $args['jumlah_kuota'],
            'status_kuota'       => $args['status_kuota']
        );
        $this->db->insert($table, $data);
        redirect('Tampilan_kuota');
    }
    function hapus_data($id)
    {
        $table = 'tbl_kuota';
        $where = array('id_kuota' => $id);
        $this->M_data->hapus_data($table, $where);
        redirect('Tampilan_kuota');
    }
    function edit_data_kuota($id)
    {
        $jb = $this->session->userdata('jb');
        $sql = $this->db->query("SELECT * FROM tbl_gunung WHERE
nama_gunung = '$jb'")->result();
        foreach ($sql as $s) {
            $idg = $s->id_gunung;
        }
        $args = $this->input->post();
        $table = 'tbl_kuota';
        $where = array('id_kuota' => $id);

```



```

        $data = array(
            'id_gunung'           => $idg,
            'tgl_kuota'           => $args['tanggal_kuota'],
            'jumlah_kuota'       => $args['jumlah_kuota'],
            'status_kuota'       => $args['status_kuota']
        );
        $this->M_data->update_data($stable, $data, $where);
        redirect('Tampilan_kuota');
    }
    public function edit_data_page($id)
    {
        //$data['level'] = $this->session->userdata('level');
        $data['content'] = 'content/halaman_edit_kuota';
        $data['title'] = 'Edit Data Kuota';
        $data['kuota'] = $this->db->query("SELECT * FROM tbl_kuota tk
JOIN tbl_gunung tg ON tg.id_gunung = tk.id_gunung WHERE tk.id_kuota =
'$id'")->result();
        //$data['kuota'] = $this->M_data->get_data_where('tbl_kuota',
array('id_kuota' => $id));
        $this->load->view('template/content',$data);
    }
    public function import(){
        $config['upload_path'] = './assets/excel';
        $config['allowed_types'] = 'xlsx|xls';
        $config['max_size'] = '1000';

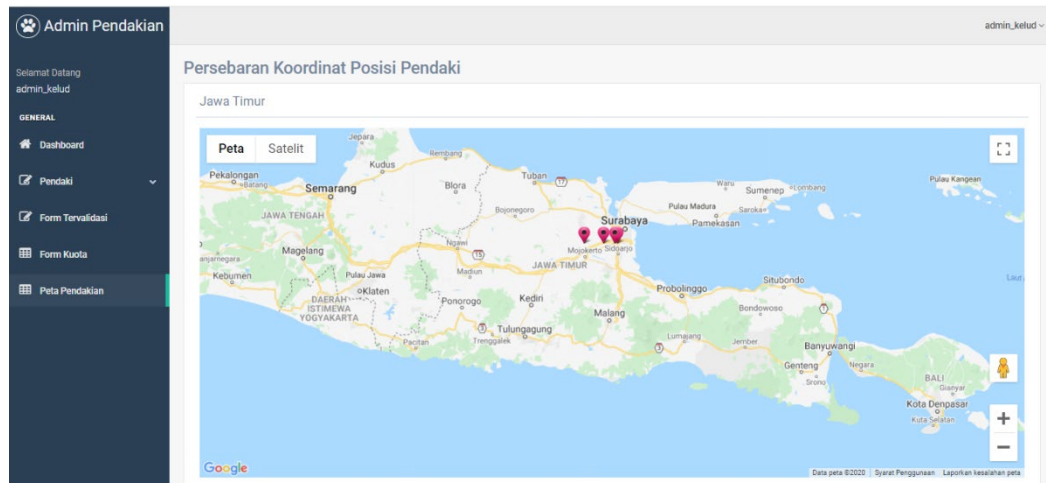
        $this->load->library('upload');
        $this->upload->initialize($config);

        if (! $this->upload->do_upload()){
            echo "gagal";
            //redirect('Tampilan_kuota');
        }else{
            $data = array('upload_data' => $this->upload->data());
            $upload_data = $this->upload->data();
            $filename = $upload_data['file_name'];
            $this->M_data->upload_kuota($filename);
            unlink('./assets/excel'.$filename);
            echo "berhasil";
            //redirect('Tampilan_kuota');
        }
    }
}
}

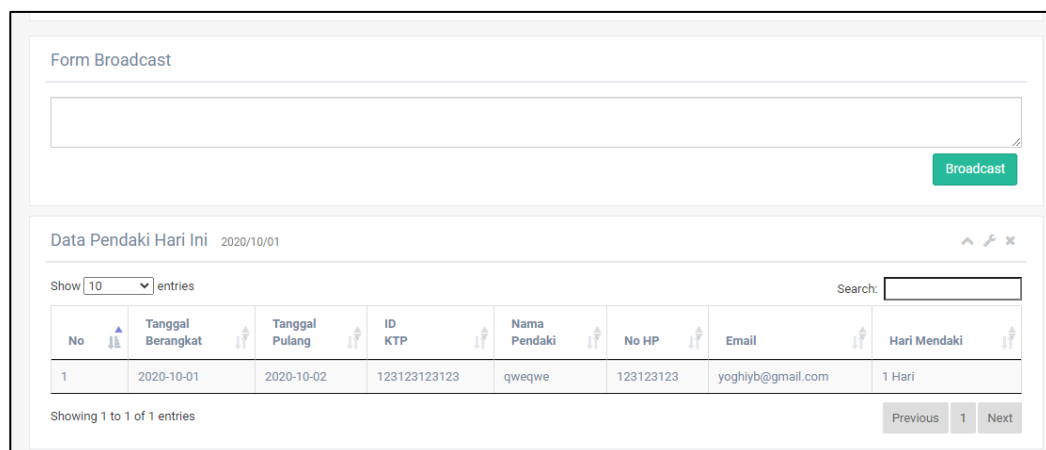
```

5.5.7. Halaman Peta Pendakian

Halaman peta pendakian merupakan halaman yang digunakan untuk menampung koordinat posisi para user sebelum melakukan pendakian, disini posisi pengguna akan direalod setiap lima menit, jika masih ada sinyal. Berikut ini desain halaman peta pendakian dari sistem ini.



Gambar 5.7. Halaman Peta Pendakian



Gambar 5.8. Halaman Form Broadcast

5.5.7.1. Coding Program Halaman Peta Pendakian

a. View Halaman Peta Pendakian

```

<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA2JUy7LvRp3vwLVJ
xf0TIYyRqSSTpo3kU&callback=initialize" async defer>
</script>
<script>
function initialize() {
    var uluru = {
        lat: -7.8039901,
        lng: 112.7380089
    };
    var map = new google.maps.Map(document.getElementById('map'), {
        zoom: 8.5,
        center: uluru
    });
    var infoWindow = new google.maps.InfoWindow;
    var bounds = new google.maps.LatLngBounds();

    function bindInfoWindow(marker, map, infoWindow, html) {
        google.maps.event.addListener(marker, 'click', function() {
            infoWindow.setContent(html);
            infoWindow.open(map, marker);
        });
    }
    function addMarker(lat, lng, info) {
        var pt = new google.maps.LatLng(lat, lng);
        bounds.extend(pt);
        var marker = new google.maps.Marker({
            map: map,
            position: pt
        });
        map.fitBounds(bounds);
        bindInfoWindow(marker, map, infoWindow, info);
    }
    <?php foreach ($peta->result() as $row) {
        $lat = $row->latitude;
        $lon = $row->longitude;
        $info = '<b>Kode Booking: '.$row->kd_booking.'\n Nama Pendaki:
'.$row->nama_pendaki.'</b>';
        echo("addMarker($lat,$lon, '$info');\n");
    } ?>

    //utk membuat data marker
    /* var features = [
    <?php //foreach ($peta->result() as $row) {
    ?> {
        position: new google.maps.LatLng(<?php //echo $row->latitude;
?>, <?php //echo $row->longitude;

```

```

        ?>),
        type: 'info',
        title: '<?php //echo "Kode Booking: " . $row->kd_booking . "\n' . "Nama
Pendaki: " . $row->nama_pendaki;
        ?>'
    },
    <?php //}
    ?>
]; */

// Create markers.
/* features.forEach(function(feature) {
    //<?php //foreach ($peta->result() as $row) {
    ?>
    var marker = new google.maps.Marker({
        position: new google.maps.LatLng(<?php //echo $row->latitude;
        ?>, <?php //echo $row->longitude;
        ?>),
        icon: "assets/merah.png",
        title: '<?php //echo "Kode Booking: " . $row->kd_booking . "\n' . "Nama
Pendaki: " . $row->nama_pendaki;
        ?>',
        map: map
    });
    <?php
    //}
    ?>
}); */
/*var marker = new google.maps.Marker({
    position: uluru,
    map: map
});*/
}
google.maps.event.addDomListener(window, 'load', initialize);
</script>
<div class="right_col" role="main">
    <div class="">
        <div class="page-title">
            <div class="title_left">
                <h3>Persebaran Koordinat Posisi Pendaki</h3>
            </div>
        </div>
        <div class="clearfix"></div>
        <div class="row">
            <div class="col-md-12 col-sm-12 col-xs-12">
                <div class="x_panel">
                    <div class="x_title">
                        <div class="row">
                            <div class="col-md-4">
                                <h2>Jawa Timur</h2>
                            </div>

```

```

<div class="col-md-4">
  <?php
  $akses = $this->session->userdata('jb');
  if ($akses == 'Admin Pusat') { ?>
    <form id="form-gunung" method="post" action="<?php echo
base_url('Peta/index/' . date("Y-m-d")); ?>">
      <select class="select2_single form-control" tabindex="-1"
id="gunung-select" name="gunung-select" required="required">
        <option value="">Pilih Gunung</option>
        <?php foreach ($gunung as $gn) { ?>
          <option value="<?= $gn->id_gunung ?>"><?=
$gn->nama_gunung ?></option>
        <?php } ?>
      </select>
      <button type="submit" style="display: none;" id="tombol-from-
gunung"></button>
    </form>
    <?php } ?>
  </div>
</div>
<div class="x_content">
  <div id="map" style="width:100%;height:500px"></div>
</div>
</div>
</div>
<div class="row">
  <div class="col-md-12 col-sm-12 col-xs-12">
    <div class="x_panel">
      <div class="x_title">
        <h2>Form Broadcast</h2>
        <div class="clearfix"></div>
      </div>
      <div class="x_content">
        <form action="<?= base_url("Peta/broadcast") ?>" method="POST">
          <textarea id="message" required="required" class="form-control"
name="message"></textarea>
          <?php
          $emails = array_column($pendakis, 'email');
          $string_emails = implode(',', $emails);
          // var_dump($string_emails);
          // $as = explode(',', $string_emails);
          // print_r($as);
          ?>
          <input type="hidden" value="<?= $string_emails ?>" name="data-email"
id="data-email">
          <button type="submit" class="btn btn-success" style="margin-top: 6px;
float: right">Broadcast</button>
        </form>
      </div>
    </div>
  </div>
</div>

```

```

</div>
</div>
</div>
<div class="row">
  <div class="col-md-12 col-sm-12 col-xs-12">
    <div class="x_panel">
      <div class="x_title">
        <h2>Data Pendaki Hari Ini <small><?= date("Y/m/d") ?></small></h2>
        <ul class="nav navbar-right panel_toolbox">
          <li><a class="collapse-link"><i class="fa fa-chevron-up"></i></a>
          </li>
          <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-expanded="false"><i class="fa fa-wrench"></i></a>
            <ul class="dropdown-menu" role="menu">
              <li><a href="#">Settings 1</a>
              </li>
              <li><a href="#">Settings 2</a>
              </li>
            </ul>
          </li>
          <li><a class="close-link"><i class="fa fa-close"></i></a>
          </li>
        </ul>
      </div>
      <div class="x_content">
        <table id="datatable-responsive" class="table table-striped table-bordered
dt-responsive nowrap" cellspacing="0" width="100%">
          <thead>
            <tr>
              <th>No</th>
              <th>Tanggal <br>Berangkat</th>
              <th>Tanggal <br>Pulang</th>
              <th>ID<br>KTP </th>
              <th>Nama<br>Pendaki</th>
              <th>No HP</th>
              <th>Email</th>
              <th>Hari Mendaki</th>
            </tr>
          </thead>
          <tbody>
            <?php
            $no = 1;
            foreach ($pendakis as $pendaki) { ?>
              <tr>
                <td><?= $no++; ?></td>
                <td><?= $pendaki->tgl_berangkat; ?></td>
                <td><?= $pendaki->tgl_pulang; ?></td>
                <td><?= $pendaki->id_ktp; ?></td>
                <td><?= $pendaki->nama_pengunjung; ?></td>

```



```

        $jb = $this->session->userdata('jb');
        $gunung = $this->input->post('gunung-select');
        $data['content'] = 'content/peta';
        $data['title'] = 'Peta Pendakian';
        $queryGunung = "SELECT t1.latitude, t1.longitude, t1.kd_booking,
t1.nama_pendaki FROM tbl_lokasi t1
                JOIN tbl_booking tb ON tb.kode_booking = t1.kd_booking
                JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung
                WHERE tg.nama_gunung = '$jb' AND tb.tgl_berangkat =
'$date'";

        if ($jb == 'Admin Pusat') {
            $queryGunung = "SELECT t1.latitude, t1.longitude,
t1.kd_booking, t1.nama_pendaki FROM tbl_lokasi t1
                JOIN tbl_booking tb ON tb.kode_booking = t1.kd_booking
                JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung
                WHERE tg.id_gunung = '$gunung' AND tb.tgl_berangkat =
'$date'";
        }

        //die(print_r($queryGunung));

        $data['peta'] = $this->db->query($queryGunung);
        $queryPendaki = "SELECT * FROM tbl_lokasi t1
                JOIN tbl_booking tb ON tb.kode_booking = t1.kd_booking
                JOIN tbl_pengunjung tp ON tp.kode_booking = tb.kode_booking
                JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung
                WHERE tb.tgl_berangkat = '$date'
                GROUP BY nama_pendaki
                ORDER BY id_lokasi DESC";

        if ($jb == 'Admin Pusat' && $gunung) {
            $queryPendaki = "SELECT * FROM tbl_lokasi t1
                JOIN tbl_booking tb ON tb.kode_booking = t1.kd_booking
                JOIN tbl_pengunjung tp ON tp.kode_booking =
tb.kode_booking
                JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung
                WHERE tb.tgl_berangkat = '$date' AND tg.id_gunung =
$gunung
                GROUP BY nama_pendaki
                ORDER BY id_lokasi DESC";
        }

        $data['pendakis'] = $this->db->query($queryPendaki)->result();

        $data['gunung'] = $this->db->query("SELECT * FROM
tbl_gunung")->result();
        // var_dump($data['peta'], $jb);
        $this->load->view('template/content', $data);
    }

```



```

public function broadcast()
{
    $message = $this->input->post('message');
    $data = $this->input->post('data-email');

    $dataArray = explode(',', $data);

    var_dump($dataArray, $data);

    foreach ($dataArray as $key => $email) {
        // PHPMailer object
        $response = false;
        $mail = new PHPMailer();

        // SMTP configuration
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'muhammadfarid031@gmail.com';
        $mail->Password = 'mARKamAH123';
        $mail->SMTPSecure = 'tls';
        $mail->Port = 587;
        $mail->SMTPDebug = SMTP::DEBUG_SERVER;

        $mail->setFrom('muhammadfarid031@gmail.com',
'Administrasi Pendaki Jawa Timur');
        $mail->addReplyTo('muhammadfarid031@gmail.com',
'Administrasi Pendaki Jawa Timur');

        // Add a recipient
        $mail->addAddress($email); //email tujuan pengiriman email

        // Email subject
        $mail->Subject = 'Informasi Cuaca';

        // Set email format to HTML
        $mail->isHTML(true);

        // Email body content
        $mailContent = $message;
        $mail->Body = $mailContent;

        // Send email
        if (!$mail->send()) {
            //echo 'Message could not be sent.';
            echo 'Mailer Error: ' . $mail->ErrorInfo;
        } else {
            //echo 'Message has been sent';
        }
        // # code...
    }
}

```


e.preventDefault

```

<?php if ($akses == 'Admin Pusat') { ?>
if (gunung.value == " || gunung.value == null) {
alert("Form gunung tidak boleh kosong!")
return
}
<?php } ?>

let iframe_place = document.getElementById('lihat-laporan')
let iframe = `<iframe width="100%" height="100%" src="<?=
base_url("laporan/laporanPendapatan/true/") ?>${date_start.value}/${date_end.val
ue}/<?= $akses == 'Admin Pusat' ? '${gunung.value}': $akses ?>"
frameborder="0"></iframe>`
iframe_place.innerHTML = "
iframe_place.innerHTML = iframe
});

$('#date-button').daterangepicker({
"showDropdowns": true,
"autoApply": true,
ranges: {
'Today': [moment(), moment()],
'Yesterday': [moment().subtract(1, 'days'), moment().subtract(1, 'days')],
'Last 7 Days': [moment().subtract(6, 'days'), moment()],
'Last 30 Days': [moment().subtract(29, 'days'), moment()],
'This Month': [moment().startOf('month'), moment().endOf('month')],
'Last Month': [moment().subtract(1, 'month').startOf('month'), moment().subtract(1,
'month').endOf('month')]
},
"locale": {
"format": "MM/DD/YYYY",
"separator": " - ",
"applyLabel": "Apply",
"cancelLabel": "Cancel",
"fromLabel": "From",
"toLabel": "To",
"customRangeLabel": "Custom",
"weekLabel": "W",
"daysOfWeek": [
"Su",
"Mo",
"Tu",
"We",
"Th",
"Fr",
"Sa"
],
"monthNames": [
"January",
"February",

```

```

"March",
"April",
"May",
"June",
"July",
"August",
"September",
"October",
"November",
"December"
],
"firstDay": 1
},
"startDate": "09/21/2020",
"endDate": "09/20/2020",
"opens": "center",
}, function(start, end, label) {
date_text.value = start.format('YYYY/MM/DD') + ' - ' +
end.format('YYYY/MM/DD')
date_start.value = start.format('YYYY-MM-DD')
date_end.value = end.format('YYYY-MM-DD')
console.log('New date range selected: ' + start.format('YYYY-MM-DD') + ' to ' +
end.format('YYYY-MM-DD') + ' (predefined range: ' + label + ')');
});
</script>

```

b. Controller Halaman Laporan Pendapatan

```

?php
defined('BASEPATH') or exit('No direct script access allowed');
class Laporan extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('M_data');
        $this->load->library('pdf');
        $this->pdf->setPaper('A4', 'potrait');
    }
    public function index()
    {
        $jb = $this->session->userdata('jb');
        $data['content'] = 'content/laporan_pembayaran';
        $data['title'] = 'Laporan Pembayarn';

        $this->load->view('template/content', $data);
    }
    public function laporanPembayaran($view = false, $start = null, $end = null)
    {

```

```

$start_date = $start == null ? $this->input->post('date-start') : $start;
$end_date = $end == null ? $this->input->post('date-end') : $end;

$query = "SELECT * FROM tbl_pengunjung tp JOIN tbl_booking tb ON
tb.kode_booking = tp.kode_booking JOIN tbl_gunung tg ON tg.id_gunung =
tb.id_gunung WHERE tb.tgl_booking BETWEEN '$start_date' AND '$end_date' AND
status_validasi = 1";

$this->pdf->filename = "laporan-data-pembayaran-" . $start_date . "-" .
$end_date . ".pdf";
$this->load->library('pdf');
$this->pdf->setPaper('A4', 'potrait');

$data["laporan"] = $this->db->query($query)->result();
$data["date_start"] = $start_date;
$data["date_end"] = $end_date;

if ($view) {
    $this->load->view("pdf/laporan_pembayaran", $data);
} else {
    $this->pdf->load_view("pdf/laporan_pembayaran", $data);
}
}

public function laporanPendapatanView()
{
    $data['content'] = 'content/laporan_pendapatan';
    $data['title'] = 'Laporan Pendapatan';

    $queryGunung = "SELECT * FROM tbl_gunung";

    $data['gunungs'] = $this->db->query($queryGunung)->result();

    $this->load->view('template/content', $data);
}

public function laporanPendapatan($view = false, $start = null, $end = null,
$gunung = null)
{
    $jb = $this->session->userdata('jb');
    $start_date = $start == null ? $this->input->post('date-start') : $start;
    $end_date = $end == null ? $this->input->post('date-end') : $end;
    $gng = $gunung == null ? $this->input->post('gunung') : $gunung;

    $queryGunung = "SELECT * FROM tbl_gunung WHERE id_gunung=" . $gng;
    if ($jb != 'Admin Pusat') {
        $queryGunung = "SELECT * FROM tbl_gunung WHERE
nama_gunung='$gng'";
    }
    $getGunung = $this->db->query($queryGunung)->row();
}

```

```

$query = "SELECT * FROM tbl_booking tb JOIN tbl_gunung tg ON
tg.id_gunung = tb.id_gunung WHERE tb.status_lunas = 1 ORDER BY
`kode_booking` DESC";
$data['laporan'] = $this->db->query($query)->result();

$data["date_start"] = $start_date;
$data["date_end"] = $end_date;
$data['gunung'] = $getGunung->nama_gunung;

if($view) {
    $this->load->view('pdf/laporan_pendapatan', $data);
} else {
    $this->pdf->load_view('pdf/laporan_pendapatan', $data);
}
}
}

```

5.5.9. Halaman Email

Halaman email ini berfungsi untuk mengirim pemberitahuan kepada user dari admin melalui email. Berikut ini desain dari halaman email.



Gambar 5.10. Halaman Email Pendaftaran



Gambar 5.11. Halaman Email Lolos Validasi



Gambar 5.12. Halaman Email Konfirmasi Pembayaran

5.5.9.1. Coding Program Halaman Email

a. Coding Halaman Email Pendaftaran

```
<?php
include 'kirimemail/PHPMailer/PHPMailerAutoload.php';

$nl = $_GET['nama'];
$email = $_GET['email'];
$kd_booking = $_GET['kd_booking'];
    $response = false;
    $mail = new PHPMailer;

    // Konfigurasi SMTP
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'muhammadfarid031@gmail.com';
    $mail->Password = 'mARKamAH123';
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    // $mail->SMTPDebug = SMTP::DEBUG_SERVER;
    $mail->setFrom('muhammadfarid031@gmail.com', 'Administrasi Pendaki
Jawa Timur');
    $mail->addReplyTo('muhammadfarid031@gmail.com', 'Administrasi
Pendaki Jawa Timur');

    // Menambahkan penerima
    $mail->addAddress($email);

    // Subjek email
    $mail->Subject = 'Informasi Booking Pendaki';

    // Mengatur format email ke HTML
```

```

$mailll->isHTML(true);

// Konten/isi email
$mailllContent = "<h1>Data Booking Anda Telah Kami Terima</h1>
<p>Hai ".$nl." terima kasih telah mendaftar di aplikasi ini, Silahkan
menunggu email kami selanjutnya ya, data anda sedang kami verifikasi.<br>
Kode booking Anda adalah: <b>".$kd_booking."</b></p>";
$mailll->Body = $mailllContent;
if(!$mailll->send()){

// echo 'Pesan tidak dapat dikirim.';
echo 'Mailer Error: ' . $mailll->ErrorInfo;
} else {
//echo 'Pesan telah terkirim';
class book {}
$response2 = new book();
$response2->success = 1;
$response2->message = "Data Masuk";
die(json_encode($response2));
}
?>

```

b. Coding Halaman Email Lolos Validasi

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
class Tampilan_validasi extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('M_data');
        require APPPATH.'libraries/phpmailer/src/Exception.php';
        require APPPATH.'libraries/phpmailer/src/PHPMailer.php';
        require APPPATH.'libraries/phpmailer/src/SMTP.php';
    }

    public function index(){
        $jb = $this->session->userdata('jb');
        $data['content'] = 'content/tampilan_validasi';
        $data['title'] = 'Form Validasi';
        //$data['pengunjung'] =
        $this->M_data->get_data_where('tbl_pengunjung', array('status_validasi' => '0'));
        $data['pengunjung'] = $this->db->query("
        SELECT *
        FROM
        tbl_pengunjung tp
        JOIN tbl_booking tb ON tb.kode_booking = tp.kode_booking
        JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung

```



```

WHERE
    tg.nama_gunung = '$jb'
    AND status_validasi = '0'';

    $this->load->view('template/content', $data);
}
public function btn_validasi($id)
{
    $table = 'tbl_pengunjung';
    $data = array('status_validasi' => '1');
    $where = array('id_pengunjung' => $id);

    $this->M_data->update_data($table,$data,$where);
    $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking WHERE tp.id_pengunjung =
'$id'")->result();

    foreach ($data as $dt) {
        $email = $dt->email;
        $nm = $dt->nama_pengunjung;
        $by = number_format($dt->biaya);
        $idb = $dt->kode_booking;
    }
    // PHPMailer object
    $response = false;
    $mail = new PHPMailer();

    // SMTP configuration
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'muhammadfarid031@gmail.com';
    $mail->Password = 'mARKamAH123';
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    $mail->setFrom('muhammadfarid031@gmail.com', 'Administrasi Pendaki Jawa
Timur');
    $mail->addReplyTo('muhammadfarid031@gmail.com', 'Administrasi Pendaki
Jawa Timur');

    // Add a recipient
    $mail->addAddress($email); //email tujuan pengiriman email

    // Email subject
    $mail->Subject = 'Informasi Booking Pendaki';

    // Set email format to HTML
    $mail->isHTML(true);

    // Email body content

```

```

    $mailContent = "<h1>Data Booking Anda Telah Lolos Validasi</h1>
    <p>Hai ".$nm." terima kasih telah menunggu, Data anda telah lolos
    validasi. Silahkan melakukan pembayaran sebesar Rp. ".$by.". Setelah melakukan
    pembayaran harap melakukan verifikasi pembayaran melalui aplikasi mobile dengan
    memasukkan kode booking anda. Kode booking anda adalah ".$idb."</p>";
    $mail->Body = $mailContent;

    // Send email
    if(!$mail->send()){
        //echo 'Message could not be sent.';
        echo 'Mailer Error: ' . $mail->ErrorInfo;
    }else{
        //echo 'Message has been sent';
    }
        redirect('Tampilan_validasi');
    }

    public function btn_hapus($id)
    {
        $stable = 'tbl_pengunjung';
        $where = array('id_pengunjung' => $id);

        $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking WHERE tp.id_pengunjung =
'$id'")->result();
        foreach ($data as $dt) {
            $email = $dt->email;
            $nm = $dt->nama_pengunjung;
            $by = number_format($dt->biaya);
            $idb = $dt->kode_booking;
        }
        // PHPMailer object
        $response = false;
        $mail = new PHPMailer();

        // SMTP configuration
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com'; //sesuaikan sesuai nama domain
        hosting/server yang digunakan
        $mail->SMTPAuth = true;
        $mail->Username = 'pendaftaranpendaki@gmail.com'; // user email
        $mail->Password = 'pendaki123'; // password email
        $mail->SMTPSecure = 'ssl';
        $mail->Port = 465;

        $mail->setFrom('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki Jawa
Timur');
        $mail->addReplyTo('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki
Jawa Timur');

```

```

// Add a recipient
$mail->addAddress($email); //email tujuan pengiriman email

// Email subject
$mail->Subject = 'Informasi Booking Pendaki';

// Set email format to HTML
$mail->isHTML(true);

// Email body content
$mailContent = "<h1>Data Booking Anda Tidak Lolos Validasi</h1>
                <p>Hai ".$nm." terima kasih telah menunggu, Mohon maaf data anda
                tidak lolos validasi. Silahkan melakukan booking ulang dengan data yang valid.</p>";
$mail->Body = $mailContent;

// Send email
if(!$mail->send()){
    //echo 'Message could not be sent.';
    //echo 'Mailer Error: ' . $mail->ErrorInfo;
}else{
    //echo 'Message has been sent';
}
$table2 = 'tbl_booking';
        $where2 = array('kode_booking' => $idb);
$this->M_data->hapus_data($table,$where);
$this->M_data->hapus_data($table2,$where2);
        redirect('Tampilan_validasi');
    }
}

```

c. Coding Halaman Email Konfirmasi Pembayaran

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
class Tampilan_form_pembayaran extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('M_data');
        require APPPATH.'libraries/phpmailer/src/Exception.php';
        require APPPATH.'libraries/phpmailer/src/PHPMailer.php';
        require APPPATH.'libraries/phpmailer/src/SMTP.php';
    }

    public function index(){
        $jb = $this->session->userdata('jb');
        $data['content'] = 'content/tampilan_form_pembayaran';
        $data['title'] = 'Form Pembayaran';
    }
}

```

```

        // $data['konfirmasi'] = $this->M_data->get_data('tbl_konfirmasi');
        $data['konfirmasi'] = $this->db->query("SELECT * FROM
tbl_konfirmasi tk JOIN tbl_booking tb ON tb.kode_booking = tk.kode_booking
JOIN tbl_gunung tg ON tg.id_gunung = tb.id_gunung WHERE tg.nama_gunung =
'$jb' AND tk.status_konfirmasi = '0'");

        $this->load->view('template/content', $data);
    }
    public function btn_validasi($id)
    {
        $table = 'tbl_konfirmasi';
        $data = array('status_konfirmasi' => '1');
        $where = array('id_konfirmasi' => $id);

        $this->M_data->update_data($table,$data,$where);
        $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking JOIN tbl_konfirmasi tk
ON tk.kode_booking = tb.kode_booking WHERE tk.id_konfirmasi =
'$id'")->result();
        foreach ($data as $dt) {
            $email = $dt->email;
            $nm = $dt->nama_pengunjung;
            $by = number_format($dt->biaya);
            $idb = $dt->kode_booking;
        }
        // PHPMailer object
        $response = false;
        $mail = new PHPMailer();

        // SMTP configuration
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'muhammadfarid031@gmail.com';
        $mail->Password = 'mARKamAH123';
        $mail->SMTPSecure = 'tls';
        $mail->Port = 587;

        $mail->setFrom('muhammadfarid031@gmail.com', 'Administrasi Pendaki
Jawa Timur');
        $mail->addReplyTo('muhammadfarid031@gmail.com', 'Administrasi
Pendaki Jawa Timur');

        // Add a recipient
        $mail->addAddress($email); //email tujuan pengiriman email

        // Email subject
        $mail->Subject = 'Informasi Booking Pendaki';

        // Set email format to HTML

```

```

$mail->isHTML(true);

// Email body content
$mailContent = "<h1>Data Booking Anda Telah Terbayar</h1>
                <p>Hai ".$nm." terima kasih telah mengikuti prosedur, Data anda
                telah lolos validasi dan telah terbayar. Silahkan menunjukkan email ini sebagai
                bukti bahwa anda telah mendaftar secara online. Selamat Mendaki, Keep safety
                and be fun.</p>";
$mail->Body = $mailContent;

// Send email
if(!$mail->send()){
    //echo 'Message could not be sent.';
    //echo 'Mailer Error: ' . $mail->ErrorInfo;
}else{
    //echo 'Message has been sent'; }
$table1 = 'tbl_booking';
$data1 = array('status_lunas' => '1');
$where1 = array('kode_booking' => $idb);

$this->M_data->update_data($table1,$data1,$where1);

$table2 = 'tbl_pengunjung';
$data2 = array('status_pembayaran' => '1');
$where2 = array('kode_booking' => $idb);

$this->M_data->update_data($table2,$data2,$where2);
redirect("Tampilan_form_pembayaran");}

public function btn_hapus($id)
{
    $data = $this->db->query("SELECT * FROM tbl_pengunjung tp JOIN
tbl_booking tb ON tb.kode_booking = tp.kode_booking JOIN tbl_konfirmasi tk
ON tk.kode_booking = tb.kode_booking WHERE tk.id_konfirmasi =
'$id'")->result();
    foreach ($data as $dt) {
        $email = $dt->email;
        $nm = $dt->nama_pengunjung;
        $by = number_format($dt->biaya);
        $idb = $dt->kode_booking;}

    // PHPMailer object
    $response = false;
    $mail = new PHPMailer();

    // SMTP configuration
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com'; //sesuaikan sesuai nama domain
    hosting/server yang digunakan
    $mail->SMTPAuth = true;
    $mail->Username = 'pendaftaranpendaki@gmail.com'; // user email
    $mail->Password = 'pendaki123'; // password email

```

```

$mail->SMTPSecure = 'ssl';
$mail->Port      = 465;

$mail->setFrom('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki
Jawa Timur');
$mail->addReplyTo('pendaftaranpendaki@gmail.com', 'Administrasi Pendaki
Jawa Timur');
// Add a recipient
$mail->addAddress($mail); //email tujuan pengiriman email

// Email subject
$mail->Subject = 'Informasi Booking Pendaki';

// Set email format to HTML
$mail->isHTML(true);

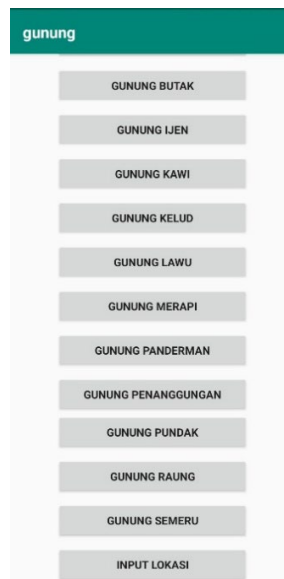
// Email body content
$mailContent = "<h1>Data Booking Anda Tidak Terbayara</h1>
                <p>Hai ".$nm." terima kasih telah mengikuti prosedur, Data booking
tidak terbayar. Silahkan melakukan booking ulang dengan data yang valid.</p>";
$mail->Body = $mailContent;
// Send email
if(!$mail->send()){
    //echo 'Message could not be sent.';
    //echo 'Mailer Error: ' . $mail->ErrorInfo;
}else{
    //echo 'Message has been sent';
}
}
$stable = 'tbl_konfirmasi';
$where = array('id_konfirmasi' => $id);
$stable1 = 'tbl_booking';
    $where1 = array('kode_booking' => $idb);

    $stable2 = 'tbl_pengunjung';
    $where2 = array('kode_booking' => $idb);
    $this->M_data->hapus_data($stable,$where);
    $this->M_data->hapus_data($stable2,$where2);
    $this->M_data->hapus_data($stable1,$where1);
    redirect("Tampilan_form_pembayaran");} }

```

5.5.10. Halaman Dashboard Android

Halaman Dashboard Adnroid berfungsi untuk memilih gunung yang akan di booking oleh para user. Di halaman ini, pengguna hanya memilih satu gunung pendakian. Berikut ini desain dari halaman pencarian toko.



Gambar 5.13. Halaman Dashboard Android

5.5.10.1. Coding Program Halaman Dashboard Android

```

package asus.com.gunung;

import...

public class MainMenuActivity extends AppCompatActivity implements
View.OnClickListener,
    GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener,
    android.location.LocationListener,
    OnMapReadyCallback {
    Button
    tombolargopuro, tombolarjuno, tombolbaluran, tombolbromo, tombolbutak, to
    mbolijen, tombolkawi, tombolkelud,

    tombollawu, tombolmerapi, tombolpanderman, tombolpenanggungan, tombolpun
    dak, tombolraung, tombolsemeru, tombolwelirang,
        tombolwidodoren, btnInputLokasi;

    TextView txtLat, txtLon;

    GoogleApiClient mGoogleApiClient;
    GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_menu);

        tombolargopuro      = findViewById(R.id.btn_argopuro);
        tombolarjuno         = findViewById(R.id.btn_arjuno);
        tombolbaluran       = findViewById(R.id.btn_baluran);
        tombolbromo         = findViewById(R.id.btn_bromo);
        tombolbutak         = findViewById(R.id.btn_butak);
        mbolijen            = findViewById(R.id.btn_ijen);
        tombolkawi          = findViewById(R.id.btn_kawi);
        tombolkelud         = findViewById(R.id.btn_kelud);
    }
}

```

```

        tombollawu         = findViewById(R.id.btn_lawu);
        tombolmerapi       = findViewById(R.id.btn_merapi);
        tombolpanderman    = findViewById(R.id.btn_panderman);
        tombolpenanggungan = findViewById(R.id.btn_penanggungan);
        tombolpundak       = findViewById(R.id.btn_pundak);
        tombolraung        = findViewById(R.id.btn_raung);
        tombolsemeru       = findViewById(R.id.btn_semeru);
        tombolwelirang     = findViewById(R.id.btn_welirang);
        tombolwidodoren    = findViewById(R.id.btn_widodoren);
        btnInputLokasi     = findViewById(R.id.btn_in);
        txtLat              = findViewById(R.id.textLat);
        txtLon              = findViewById(R.id.textLon);

        tombolargopuro.setOnClickListener(this);
        tombolarjuno.setOnClickListener(this);
        tombolbaluran.setOnClickListener(this);
        tombolbromo.setOnClickListener(this);
        tombolbutak.setOnClickListener(this);
        tombolijen.setOnClickListener(this);
        tombolkawi.setOnClickListener(this);
        tombolkelud.setOnClickListener(this);
        tombollawu.setOnClickListener(this);
        tombolmerapi.setOnClickListener(this);
        tombolpanderman.setOnClickListener(this);
        tombolpenanggungan.setOnClickListener(this);
        tombolpundak.setOnClickListener(this);
        tombolraung.setOnClickListener(this);
        tombolsemeru.setOnClickListener(this);
        tombolwelirang.setOnClickListener(this);
        tombolwidodoren.setOnClickListener(this);
        btnInputLokasi.setOnClickListener(this);

        GPSTracker mGPS = new GPSTracker(this);
        if (mGPS.canGetLocation()) {
            mGPS.getLocation();
            txtLat.setText(String.valueOf(mGPS.getLatitude()));
            txtLon.setText(String.valueOf(mGPS.getLongitude()));
        } else {
            Toast.makeText(this, "Unable to find",
                Toast.LENGTH_LONG).show();
            System.out.println("Unable");
        }

        android.location.LocationListener locListener = new
        MainMenuActivity();
        //Mengolah data lokasi menggunakan GPS
        LocationManager lm = (LocationManager)
        getSystemService(Context.LOCATION_SERVICE);
        if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED
            && ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            return;
        }
        assert lm != null;
        lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 100,
        1, locListener);
    }

```



```

protected void buildGoogleApiClient() {
    mGoogleApiClient = new
GoogleApiClient.Builder(this).addConnectionCallbacks(this).addOnConn
ectionFailedListener(this).addApi(LocationServices.API).build();
    mGoogleApiClient.connect();
}

@Override
public void onClick(View v) {
    if (v == tombolargopuro) {
        startActivity(new Intent(this,
DetailGunungArgopuroActivity.class));
    }
    if (v == tombolarjuno) {
        startActivity(new Intent(this,
DetailGunungArjunoActivity.class));
    }
    if (v == tombolbaluran) {
        startActivity(new Intent(this,
DetailGunungBaluranActivity.class));
    }
    if (v == tombolbromo) {
        startActivity(new Intent(this,
DetailGunungBromoActivity.class));
    }
    if (v == tombolbutak) {
        startActivity(new Intent(this,
DetailGunungButakActivity.class));
    }
    if (v == tombolijen) {
        startActivity(new
Intent(this, DetailGunungIjenActivity.class));
    }
    if (v == tombolkawi) {
        startActivity(new Intent(this,
DetailGunungKawiActivity.class));
    }
    if (v == tombolkelud) {
        startActivity(new Intent(this,
DetailGunungKeludActivity.class));
    }
    if (v == tombollawu) {
        startActivity(new Intent(this,
DetailGunungLawuActivity.class));
    }
    if (v == tombolmerapi) {
        startActivity(new Intent(this,
DetailGunungMerapiActivity.class));
    }
    if (v == tombolpanderman) {
        startActivity(new Intent(this,
DetailGunungPandermanActivity.class));
    }
    if (v == tombolpenanggungan) {
        startActivity(new Intent(this,
DetailGunungPenanggunganActivity.class));
    }
    if (v == tombolpundak) {
        startActivity(new Intent(this,
DetailGunungPundakActivity.class));
    }
    if (v == tombolraung) {
        startActivity(new Intent(this,

```

```

DetailGunungRaungActivity.class));
    }
    if (v == tombolsemeru) {
        startActivity(new Intent(this,
DetailGunungSemeruActivity.class));
    }
    if (v == tombolwelirang) {
        startActivity(new Intent(this,
DetailGunungWelirangActivity.class));
    }
    if (v == tombolwidodoren) {
        startActivity(new Intent(this,
DetailGunungWidodorenActivity.class));
    }
    if (v == btnInputLokasi) {
        Intent intent = null;
        intent = new Intent(this, InputLokasiActivity.class);
        Bundle bundle = new Bundle();
        bundle.putString("latitude",
String.valueOf(txtLat.getText()));
        bundle.putString("longitude",
String.valueOf(txtLon.getText()));
        intent.putExtras(bundle);
        startActivity(intent);
    }
}

@Override
public void onConnected(@Nullable Bundle bundle) {
    Toast.makeText(this, "Lokasi terdeteksi",
Toast.LENGTH_LONG).show();
    LocationRequest mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(1000);
    mLocationRequest.setFastestInterval(1000);

    mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER
_ACCURACY);
    if (ContextCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {

        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient
lient, mLocationRequest, (LocationListener) this);
    }
}

@Override
public void onConnectionSuspended(int i) {
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult
connectionResult) {
}

@Override
public void onLocationChanged(Location location) {

    Location mLastLocation = location;

    //menghentikan pembaruan lokasi
    if (mGoogleApiClient != null) {

```

```

LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient, (LocationListener) this);
    }
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override
public void onProviderEnabled(String provider) {

}

@Override
public void onProviderDisabled(String provider) {

}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    //Memulai Google Play Services
    if (android.os.Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
            buildGoogleApiClient();
            mMap.setMyLocationEnabled(true);
            Toast.makeText(this, "Lokasi berubah",
Toast.LENGTH_LONG).show();
        }
    } else {
        buildGoogleApiClient();
        mMap.setMyLocationEnabled(true);
    }
}
}
}

```

5.5.11. Halaman Utama Android

Halaman utama android dalam sistem ini adalah suatu halaman yang berguna untuk mengakses keseluruhan menu dalam sistem ini. Berikut ini desain halaman utama dari sistem ini.



Gambar 5.14. Halaman Utama Android

5.5.11.1. Coding Program Halaman Utama Android

Coding ini ada di semua halaman utama gunung. Disini hanya mengambil salah satu coding dari semua halaman utama gunung.

```

package asus.com.gunung.detail_gunung;

import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import asus.com.gunung.MainActivity;
import asus.com.gunung.R;
import asus.com.gunung.booking_gunung.BookingGunungActivity;
import asus.com.gunung.booking_gunung.InformasiGunungActivity;
import asus.com.gunung.booking_gunung.KuotaGunungActivity;
import asus.com.gunung.booking_gunung.Search_Konfirmasi_Bayar;

public class DetailGunungArgopuroActivity extends AppCompatActivity
implements View.OnClickListener {
    Button tombolinformasi, tombolkuota, tombolbooking,
    tombolpencarian;

    public final static String TAG_NAMAG = "namag";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_gunung_argopuro);

        tombolinformasi = findViewById(R.id.btn_infor_argopuro);
    }
}

```

```

        tombolkuota      =findViewById(R.id.btn_kuota_argopuro);
        tombolbooking
=findViewById(R.id.btn_booking_argopuro);
        tombolpencarian
=findViewById(R.id.btn_pencarian_argopuro);

        tombolinformasi.setOnClickListener(this);
        tombolkuota.setOnClickListener(this);
        tombolbooking.setOnClickListener(this);
        tombolpencarian.setOnClickListener(this);
    }

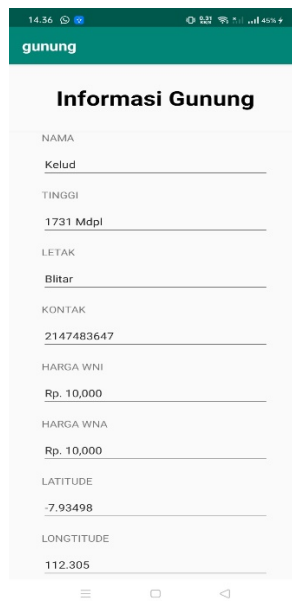
    @Override
    public void onClick(View v) {

        if (v == tombolinformasi) {
            Intent intent = new
Intent(DetailGunungArgopuroActivity.this,
InformasiGunungActivity.class);
            String namag = "Argapura";
            intent.putExtra(TAG_NAMAG, namag);
            finish();
            startActivity(intent);
        }
        if (v == tombolkuota) {
            Intent intent = new
Intent(DetailGunungArgopuroActivity.this,
KuotaGunungActivity.class);
            String namag = "Argapura";
            intent.putExtra(TAG_NAMAG, namag);
            finish();
            startActivity(intent);
        }
        if (v == tombolbooking) {
            Intent intent = new
Intent(DetailGunungArgopuroActivity.this,
BookingGunungActivity.class);
            String namag = "Argapura";
            intent.putExtra(TAG_NAMAG, namag);
            finish();
            startActivity(intent);
        }
        if (v == tombolpencarian) {
            startActivity(new Intent(this,
Search_Konfirmasi_Bayar.class));
        }
    }
}

```

5.5.12. Halaman Informasi Gunung Android

Halaman informasi gunung ini berfungsi untuk menginformasikan data gunung pendakian. Jadi sangat berguna bagi para user baru yang mendaki gunung. Berikut ini desain dari halaman informasi gunung.



Gambar 5.15. Halaman Informasi Gunung Android

5.5.12.1. Coding Program Halaman Informasi Gunung Android

Coding ini ada di semua halaman informasi gunung. Disini hanya mengambil salah satu coding dari semua halaman informasi gunung yang tersedia di halaman android.

```

package asus.com.gunung.booking_gunung;

import..

public class InformasiGunungActivity extends AppCompatActivity {
    int success;
    TextView txtnama, txttinggi, txtletak, txtkontak, txthwni,
    txthwna, txtlatitude, txtlongitude;
    public static final String TAG_NAMA = "nama";
    public static final String TAG_LETAK = "letak";
    public static final String TAG_TINGGI = "tinggi";
    public static final String TAG_KONTAK = "kontak";
    public static final String TAG_HWNI = "hwni";
    public static final String TAG_HWNA = "hwna";
    public static final String TAG_LATITUDE = "latitude";
    public static final String TAG_LONGTITUDE = "longitude";
    public static final String TAG_NAMAG = "namag";
    public static final String TAG_SUCCESS = "success";
    public static final String TAG_MESSAGE = "message";
    String tag_json_obj = "json_obj_req";
    ConnectivityManager conMgr;
    SharedPreferences sharedpreferences;
    String namag;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_informasi_gunung);
        conMgr = (ConnectivityManager)

```

```

getSystemService(Context.CONNECTIVITY_SERVICE);
{
    if (conMgr.getActiveNetworkInfo() != null
        && conMgr.getActiveNetworkInfo().isAvailable()
        && conMgr.getActiveNetworkInfo().isConnected())
    {
    } else {
        Toast.makeText(getApplicationContext(), "No Internet
Connection",
            Toast.LENGTH_LONG).show();
    }
}
txtnama = (TextView) findViewById(R.id.nama);
txtletak = (TextView) findViewById(R.id.letak);
txttinggi = (TextView) findViewById(R.id.tinggi);
txtkontak = (TextView) findViewById(R.id.kontak);
txthwni = (TextView) findViewById(R.id.hwni);
txthwna = (TextView) findViewById(R.id.hwna);
txtlatitude = (TextView) findViewById(R.id.latitude);
txtlongtitude = (TextView) findViewById(R.id.longtitude);
namag = getIntent().getStringExtra(TAG_NAMAG);
getdata(namag);
}
private void getdata(final String namag) {
    StringRequest strReq = new
StringRequest(Request.Method.POST, Server.URL + "get_info.php", new
Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        JSONObject j = null;
        try {
            JSONObject jsonObj = new JSONObject(response);
            success = jsonObj.getInt(TAG_SUCCESS);
            if (success == 1) {
                String nm = jsonObj.getString(TAG_NAMA);
                String lt = jsonObj.getString(TAG_LETAK);
                String tg = jsonObj.getString(TAG_TINGGI);
                String kt = jsonObj.getString(TAG_KONTAK);
                String hi = jsonObj.getString(TAG_HWNI);
                String ha = jsonObj.getString(TAG_HWNA);
                String la = jsonObj.getString(TAG_LATITUDE);
                String lo = jsonObj.getString(TAG_LONGTITUDE);

                Toast.makeText(getApplicationContext(),
jObj.getString(TAG_MESSAGE), Toast.LENGTH_LONG).show();

                txtnama.setText(nm);
                txtletak.setText(lt);
                txttinggi.setText(tg);
                txtkontak.setText(kt);
                txthwni.setText(hi);
                txthwna.setText(ha);
                txtlatitude.setText(la);
                txtlongtitude.setText(lo);
            } else {
                Toast.makeText(getApplicationContext(),
                    jsonObj.getString(TAG_MESSAGE),
Toast.LENGTH_LONG).show();
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}

```

```

    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    }) {

    @Override
    protected Map<String, String> getParams() {
        // Posting parameters to login url
        Map<String, String> params = new HashMap<String,
String>();
        params.put("nama", namag);

        return params;
    }
};
RequestQueue requestQueue = Volley.newRequestQueue(this);
requestQueue.add(strReq);
AppController.getInstance().addToRequestQueue(strReq,
tag_json_obj);
}
}

```

5.5.13. Halaman Kuota Gunung Android

Halaman kuota gunung ini berfungsi untuk informasi data kuota gunung pendakian. Jadi di halaman kuota gunung ini, para user dapat melihat jumlah kuota gunung yang disediakan oleh admin. Berikut ini desain dari halaman kuota gunung.



Gambar 5.16. Halaman Kuota Gunung Android

5.5.13.1. Coding Program Halaman Kuota Gunung Android

Coding ini ada di semua halaman kuota gunung. Disini hanya mengambil salah satu coding dari semua halaman kuota gunung yang disediakan sistem.


```

package asus.com.gunung.booking_gunung;

public class ListKuota {
    private String tanggal;
    private String jumlah;
    private String status;

    public void setTanggal(String tanggal) {
        this.tanggal = tanggal;
    }
    String getTanggal() {
        return tanggal;
    }

    public void setJumlah (String jumlah) {
        this.jumlah = jumlah;
    }
    public String getJumlah() {
        return jumlah;
    }

    public void setStatus (String status) {
        this.status = status;
    }
    String getStatus() {
        return status; }}

```

a. ListKuotaAdapter.java

Fungsi dari ListKuotaAdapter.java sendiri berfungsi menampung data dari API ke list kuota yang tersedia. Berikut coding dari ListKuotaAdapter.java.

```

package asus.com.gunung.booking_gunung;

import...

public class ListKuotaAdapter extends BaseAdapter {
    //private ArrayList<HashMap<String, String>> data;
    private ArrayList<ListKuota> list_kuota = new
ArrayList<ListKuota>();
    private static LayoutInflater inflater = null;

    public ListKuotaAdapter(Activity a, ArrayList<ListKuota> d) {
        list_kuota = d;
        inflater = (LayoutInflater)
a.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    public int getCount() {
        return list_kuota.size();
    }

    public Object getItem(int position) {
        return list_kuota.get(position);
    }

    public long getItemId(int position) {
        return position;
    }

    @SuppressWarnings("InflateParams")

```

```

@Override
public View getView(int position, View convertView, ViewGroup
parent) {
    View vi = convertView;
    if (convertView == null) {
        vi = inflater.inflate(R.layout.listview_kuota, null);
    }
    TextView tanggal = vi.findViewById(R.id.Tanggal);
    TextView kuota = vi.findViewById(R.id.JumlahKuota);
    TextView status = vi.findViewById(R.id.StatusGunung);

    ListKuota daftar_ponpes = list_kuota.get(position);
    tanggal.setText(daftar_ponpes.getTanggal());
    kuota.setText(daftar_ponpes.getJumlah());
    status.setText(daftar_ponpes.getStatus());
    return vi;
}

public void setFilter(List<ListKuota> newList){
    list_kuota = new ArrayList<>();
    list_kuota.addAll(newList);
    notifyDataSetChanged();
}
}

```

b. KuotaGunungActivity.java

KuotaGunungActivity.java berfungsi untuk melihat kuota gunung yang disediakan oleh admin. Berikut coding dari KuotaGunungActivity.java.

```

package asus.com.gunung.booking_gunung;

import...

public class KuotaGunungActivity extends AppCompatActivity implements
TextWatcher {

    ListView list;
    JSONParser jParser = new JSONParser();
    ArrayList<ListKuota> list_kuota = new ArrayList<ListKuota>();
    JSONArray listKuota, daftarSearchKuota = null;
    ListKuotaAdapter adapter;
    EditText Query;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kuota_gunung);

        list = findViewById(R.id.ListViewKuota);
        Query = findViewById(R.id.EditTxtCariKuota);

        Query.addTextChangedListener(this);

        //jalankan ReadDataTask
        ReadDataTask m= (ReadDataTask) new ReadDataTask().execute();
    }
}

```

```

@SuppressLint("StaticFieldLeak")
class ReadDataTask extends AsyncTask<String, String, String> {
    ProgressDialog pDialog;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(KuotaGunungActivity.this);
        pDialog.setTitle("Mengambil Daftar Data Kuota");
        pDialog.setMessage("Tunggu Sebentar");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    @Override
    protected String doInBackground(String... sText) {
        ListKuota tempKuota = new ListKuota();
        List<NameValuePair> parameter = new
ArrayList<NameValuePair>();
        parameter.add(new BasicNameValuePair("namag",
getIntent().getStringExtra("namag")));
        try {
            JSONObject json = jParser.makeHttpRequest(Server.URL +
"kuota.php", "POST", parameter);
            int success = json.getInt("success");

            //Toast.makeText(ListTokoActivity.this, "JSON = " +
json, Toast.LENGTH_LONG).show();
            if (success == 1 ) { //Ada record Data (SUCCESS = 1)
                //Getting Array of daftar_mhs
                listKuota = json.getJSONArray("kuota");
                // looping through All daftar_mhs
                for (int i = 0; i < listKuota.length(); i++){

                    JSONObject c = listKuota.getJSONObject(i);
                    tempKuota = new ListKuota();
                    tempKuota.setTanggal(c.getString("tanggal"));
                    tempKuota.setJumlah(c.getString("jumlah"));
                    tempKuota.setStatus(c.getString("status"));
                    list_kuota.add(tempKuota);
                }
                return "OK";
            }else{
                //Tidak Ada Record Data (SUCCESS = 0)
                return "no results";
            }
        } catch (Exception e) {
            e.printStackTrace();
            return "Exception Caught";
        }
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        pDialog.dismiss();
        if(result.equalsIgnoreCase("Exception Caught")){
            Toast.makeText(KuotaGunungActivity.this, "Unable to
connect to server,please check your internet connection!",
Toast.LENGTH_LONG).show();
        }
        if(result.equalsIgnoreCase("no results")){

```

```

        Toast.makeText(KuotaGunungActivity.this, "Data empty",
Toast.LENGTH_LONG).show();
    }else {
        //mengambil adatper data toko
        adapter = new
ListKuotaAdapter(KuotaGunungActivity.this, list_kuota);
        list.setAdapter(adapter);
    }
}

@Override
public void beforeTextChanged(CharSequence s, int start, int count,
int after) {

}

@Override
public void onTextChanged(CharSequence s, int start, int before,
int count) {

}

@Override
public void afterTextChanged(Editable s) {
    String names = s.toString();
    //String alamat = s.toString();
    List<ListKuota> newList = new ArrayList<>();

    for (ListKuota r : list_kuota) {
        //ListToko akan mereplace list_toko
        String tanggal = r.getTanggal();
        //String alamat1 = r.getAlamat();
        //|| alamat1.toLowerCase().contains(alamat.toLowerCase())
        if (tanggal.toLowerCase().contains(names.toLowerCase())) {
            newList.add(r);
        }
    }

    //make seFilter method in adapter and pass your new list
    adapter.setFilter(newList);
}
}

```

5.5.14. Halaman Booking Gunung Android

Halaman Booking gunung ini berfungsi untuk mendaftar pendakian. Halaman ini termasuk halaman yang penting, karena disini dimulainya para user mengisi data diri dan megirimkannya ke admin. Berikut ini desain dari halaman booking gunung.

Gambar 5.17. Halaman Booking Gunung Android

5.5.14.1. Coding Program Halaman Booking Gunung Android

Coding ini ada di semua halaman booking gunung. Disini hanya mengambil salah satu coding dari semua halaman booking gunung yang disediakan sistem.

```

package asus.com.gunung.booking_gunung;
import...

public class BookingGunungActivity extends AppCompatActivity implements
View.OnClickListener {

    EditText tgl_berangkat, tgl_pulang, id_pengunjung, id_ktp,
id_ktp_leader, nama_lengkap, tanggal_lahir, alamat_lengkap, kebangsaan,
umur, email, no_hp, no_hp_ortu, pekerjaan, status_porter;
    EditText id_ktp1, nama1, telp1, id_ktp2, nama2, telp2, id_ktp3,
nama3, telp3, id_ktp4, nama4, telp4, id_ktp5, nama5, telp5,
id_ktp6, nama6, telp6, id_ktp7, nama7, telp7, id_ktp8,
nama8, telp8, id_ktp9, nama9, telp9, id_ktp10, nama10, telp10;
    Spinner warga1, jk1, warga2, jk2, warga3, jk3, warga4, jk4,
warga5, jk5, warga6, jk6, warga7, jk7, warga8, jk8, warga9, jk9,
warga10, jk10, status_leader;
    String NamaGunung, TglBerangkat, TglPulang, idPengunjung, idKtp,
statusLeader, idKtpLeader, namaLengkap, tanggalLahir, alamatLengkap,
Kebangsaan, Umur,
Email, noHp, noHpOrtu, Pekerjaan, statusPorter;
    String IdKtp1, Nama1, Telp1, Warga1, Jk1, IdKtp2, Nama2, Telp2,
Warga2, Jk2, IdKtp3, Nama3, Telp3, Warga3, Jk3, IdKtp4, Nama4, Telp4,
Warga4, Jk4,
IdKtp5, Nama5, Telp5, Warga5, Jk5, IdKtp6, Nama6, Telp6,
Warga6, Jk6, IdKtp7, Nama7, Telp7, Warga7, Jk7, IdKtp8, Nama8, Telp8,

```

```

Warga8, Jk8,
    IdKtp9, Nama9, Telp9, Warga9, Jk9, IdKtp10, Nama10, Telp10,
Warga10, Jk10;
    Button button_simpan;

    private SimpleDateFormat dateFormatter;
    private TextView tvDateResult;
    private Button btDatePicker;
    private DatePickerDialog datePickerDialog;
    private TextView tvDateResult1;
    private Button btDatePicker1;
    public static final String TAG_NAMAG = "namag";
    public static final String TAG_SUCCESS = "success";
    public static final String TAG_MESSAGE = "message";
    String namag;
    int success;
    String tag_json_obj = "json_obj_req";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_booking_gunung);

        dateFormatter = new SimpleDateFormat("dd-MM-yyyy", Locale.US);

        tvDateResult = (TextView)
findViewById(R.id.edit_text_tgl_berangkat);
        btDatePicker = (Button) findViewById(R.id.bt_date_berangkat);
        btDatePicker.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showDateDialog();
            }
        });
        tvDateResult1 = (TextView)
findViewById(R.id.edit_text_tgl_pulang);
        btDatePicker1 = (Button) findViewById(R.id.bt_date_pulang);
        btDatePicker1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showDateDialog1();
            }
        });
        namag = getIntent().getStringExtra(TAG_NAMAG);

        id_ktp
            = findViewById(R.id.edit_text_id_ktp);
        status_leader
            =
findViewById(R.id.edit_text_status_leader);
        id_ktp_leader
            =
findViewById(R.id.edit_text_id_ktp_leader);
        nama_lengkap
            =
findViewById(R.id.edit_text_nama_lengkap);
        tanggal_lahir
            = findViewById(R.id.edit_text_tgl_lahir);
        alamat_lengkap
            =
findViewById(R.id.edit_text_alamat_lengkap);
        kebangsaan
            = findViewById(R.id.edit_text_kebangsaan);
        umur
            = findViewById(R.id.edit_text_umur);
        email
            = findViewById(R.id.edit_text_email);
        no_hp
            = findViewById(R.id.edit_text_no_hp);
        no_hp_ortu
            = findViewById(R.id.edit_text_no_hp_ortu);
        pekerjaan
            = findViewById(R.id.edit_text_pekerjaan);
        status_porter
            =
findViewById(R.id.edit_text_status_porter);

        id_ktp1 = findViewById(R.id.edtNoIdenAng1);

```

```

    nama1 = findViewById(R.id.edtNamaAng1);
    telp1 = findViewById(R.id.edtNoTelpAng1);
    warga1 = findViewById(R.id.spinnerKWAng1);
    jk1 = findViewById(R.id.spinnerJKAng1);

    id_ktp2 = findViewById(R.id.edtNoIdenAng2);
    nama2 = findViewById(R.id.edtNamaAng2);
    telp2 = findViewById(R.id.edtNoTelpAng2);
    warga2 = findViewById(R.id.spinnerKWAng2);
    jk2 = findViewById(R.id.spinnerJKAng2);

    id_ktp3 = findViewById(R.id.edtNoIdenAng3);
    nama3 = findViewById(R.id.edtNamaAng3);
    telp3 = findViewById(R.id.edtNoTelpAng3);
    warga3 = findViewById(R.id.spinnerKWAng3);
    jk3 = findViewById(R.id.spinnerJKAng3);

    id_ktp4 = findViewById(R.id.edtNoIdenAng4);
    nama4 = findViewById(R.id.edtNamaAng4);
    telp4 = findViewById(R.id.edtNoTelpAng4);
    warga4 = findViewById(R.id.spinnerKWAng4);
    jk4 = findViewById(R.id.spinnerJKAng4);

    id_ktp5 = findViewById(R.id.edtNoIdenAng5);
    nama5 = findViewById(R.id.edtNamaAng5);
    telp5 = findViewById(R.id.edtNoTelpAng5);
    warga5 = findViewById(R.id.spinnerKWAng5);
    jk5 = findViewById(R.id.spinnerJKAng5);

    id_ktp6 = findViewById(R.id.edtNoIdenAng6);
    nama6 = findViewById(R.id.edtNamaAng6);
    telp6 = findViewById(R.id.edtNoTelpAng6);
    warga6 = findViewById(R.id.spinnerKWAng6);
    jk6 = findViewById(R.id.spinnerJKAng6);

    id_ktp7 = findViewById(R.id.edtNoIdenAng7);
    nama7 = findViewById(R.id.edtNamaAng7);
    telp7 = findViewById(R.id.edtNoTelpAng7);
    warga7 = findViewById(R.id.spinnerKWAng7);
    jk7 = findViewById(R.id.spinnerJKAng7);

    id_ktp8 = findViewById(R.id.edtNoIdenAng8);
    nama8 = findViewById(R.id.edtNamaAng8);
    telp8 = findViewById(R.id.edtNoTelpAng8);
    warga8 = findViewById(R.id.spinnerKWAng8);
    jk8 = findViewById(R.id.spinnerJKAng8);

    id_ktp9 = findViewById(R.id.edtNoIdenAng9);
    nama9 = findViewById(R.id.edtNamaAng9);
    telp9 = findViewById(R.id.edtNoTelpAng9);
    warga9 = findViewById(R.id.spinnerKWAng9);
    jk9 = findViewById(R.id.spinnerJKAng9);

    id_ktp10 = findViewById(R.id.edtNoIdenAng10);
    nama10 = findViewById(R.id.edtNamaAng10);
    telp10 = findViewById(R.id.edtNoTelpAng10);
    warga10 = findViewById(R.id.spinnerKWAng10);
    jk10 = findViewById(R.id.spinnerJKAng10);

    button_simpan = findViewById(R.id.btn_simpan);
    button_simpan.setOnClickListener(this);
}

private void showDateDialog1() {
    Calendar newCalendar = Calendar.getInstance();

```

```

        datePickerDialog = new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {

            @Override
            public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {

                Calendar newDate = Calendar.getInstance();
                newDate.set(year, monthOfYear, dayOfMonth);

tvDateResult1.setText(dateFormatter.format(newDate.getTime()));

                }, newCalendar.get(Calendar.YEAR),
newCalendar.get(Calendar.MONTH),
newCalendar.get(Calendar.DAY_OF_MONTH));

            datePickerDialog.show();
        }

        private void showDateDialog() {
            Calendar newCalendar = Calendar.getInstance();
            datePickerDialog = new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {

                @Override
                public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {

                    Calendar newDate = Calendar.getInstance();
                    newDate.set(year, monthOfYear, dayOfMonth);

tvDateResult.setText(dateFormatter.format(newDate.getTime()));

                }

                }, newCalendar.get(Calendar.YEAR),
newCalendar.get(Calendar.MONTH),
newCalendar.get(Calendar.DAY_OF_MONTH));
            datePickerDialog.show();
        }

        @Override
        public void onClick(View v) {
            if(v == button_simpan){
                new addData().execute();
                Toast.makeText(BookingGunungActivity.this, "Input
Berhasil", Toast.LENGTH_SHORT).show();
                startActivity(new Intent(this, MainMenuActivity.class));
            }
        }

        @SuppressWarnings("StaticFieldLeak")
        class addData extends AsyncTask<Void, Void, String> {

            ProgressDialog pDialog;

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                pDialog = new ProgressDialog(BookingGunungActivity.this);
                pDialog.setMessage("Menambahkan Data");
                pDialog.setIndeterminate(true);
                pDialog.setCancelable(true);
                pDialog.show();
            }
        }
    }

```



```

@Override
protected String doInBackground(Void... params) {

   >NamaGunung = getIntent().getStringExtra(TAG_NAMAG);
   >TglBerangkat = tvDateResult.getText().toString();
   >TglPulang = tvDateResult1.getText().toString();
   >idKtp = id_ktp.getText().toString();
   >statusLeader = status_leader.getSelectedItem().toString();
   >idKtpLeader = id_ktp_leader.getText().toString();
   >namaLengkap = nama_lengkap.getText().toString();
   >tanggalLahir = tanggal_lahir.getText().toString();
   >alamatLengkap = alamat_lengkap.getText().toString();
   >Kebangsaan = kebangsaan.getText().toString();
   >Umur = umur.getText().toString();
   >Email = email.getText().toString();
   >noHp = no_hp.getText().toString();
   >noHpOrtu = no_hp_ortu.getText().toString();
   >Pekerjaan = pekerjaan.getText().toString();
   >statusPorter = status_porter.getText().toString();

   >IdKtp1 = id_ktp1.getText().toString();
   >Nama1 = nama1.getText().toString();
   >Telp1 = telp1.getText().toString();
   >Warga1 = warga1.getSelectedItem().toString();
   >Jk1 = jk1.getSelectedItem().toString();

   >IdKtp2 = id_ktp2.getText().toString();
   >Nama2 = nama2.getText().toString();
   >Telp2 = telp2.getText().toString();
   >Warga2 = warga2.getSelectedItem().toString();
   >Jk2 = jk2.getSelectedItem().toString();

   >IdKtp3 = id_ktp3.getText().toString();
   >Nama3 = nama3.getText().toString();
   >Telp3 = telp3.getText().toString();
   >Warga3 = warga3.getSelectedItem().toString();
   >Jk3 = jk3.getSelectedItem().toString();

   >IdKtp4 = id_ktp4.getText().toString();
   >Nama4 = nama4.getText().toString();
   >Telp4 = telp4.getText().toString();
   >Warga4 = warga4.getSelectedItem().toString();
   >Jk4 = jk4.getSelectedItem().toString();

   >IdKtp5 = id_ktp5.getText().toString();
   >Nama5 = nama5.getText().toString();
   >Telp5 = telp5.getText().toString();
   >Warga5 = warga5.getSelectedItem().toString();
   >Jk5 = jk5.getSelectedItem().toString();

   >IdKtp6 = id_ktp6.getText().toString();
   >Nama6 = nama6.getText().toString();
   >Telp6 = telp6.getText().toString();
   >Warga6 = warga6.getSelectedItem().toString();
   >Jk6 = jk6.getSelectedItem().toString();

   >IdKtp7 = id_ktp7.getText().toString();
   >Nama7 = nama7.getText().toString();
   >Telp7 = telp7.getText().toString();
   >Warga7 = warga7.getSelectedItem().toString();
   >Jk7 = jk7.getSelectedItem().toString();

   >IdKtp8 = id_ktp8.getText().toString();

```

```

Nama8 = nama8.getText().toString();
Telp8 = telp8.getText().toString();
Warga8 = warga8.getSelectedItem().toString();
Jk8 = jk8.getSelectedItem().toString();

IdKtp9 = id_ktp9.getText().toString();
Nama9 = nama9.getText().toString();
Telp9 = telp9.getText().toString();
Warga9 = warga9.getSelectedItem().toString();
Jk9 = jk9.getSelectedItem().toString();

IdKtp10 = id_ktp10.getText().toString();
Nama10 = nama10.getText().toString();
Telp10 = telp10.getText().toString();
Warga10 = warga10.getSelectedItem().toString();
Jk10 = jk10.getSelectedItem().toString();

HashMap<String, String> parameter = new HashMap<String,
String>();
parameter.put("namag", NamaGunung);
parameter.put("idk", idKtp);
parameter.put("tgb", TglBerangkat);
parameter.put("tgp", TglPulang);
parameter.put("sl", statusLeader);
parameter.put("ikl", idKtpLeader);
parameter.put("nl", namaLengkap);
parameter.put("tl", tanggalLahir);
parameter.put("al", alamatLengkap);
parameter.put("kb", Kebangsaan);
parameter.put("um", Umur);
parameter.put("em", Email);
parameter.put("nh", noHp);
parameter.put("nho", noHpOrtu);
parameter.put("p", Pekerjaan);
parameter.put("sp", statusPorter);

parameter.put("nama1", Nama1);
parameter.put("no_iden1", IdKtp1);
parameter.put("no_telp1", Telp1);
parameter.put("warga1", Warga1);
parameter.put("jk1", Jk1);

parameter.put("nama2", Nama2);
parameter.put("no_iden2", IdKtp2);
parameter.put("no_telp2", Telp2);
parameter.put("warga2", Warga2);
parameter.put("jk2", Jk2);

parameter.put("nama3", Nama3);
parameter.put("no_iden3", IdKtp3);
parameter.put("no_telp3", Telp3);
parameter.put("warga3", Warga3);
parameter.put("jk3", Jk3);

parameter.put("nama4", Nama4);
parameter.put("no_iden4", IdKtp4);
parameter.put("no_telp4", Telp4);
parameter.put("warga4", Warga4);
parameter.put("jk4", Jk4);

parameter.put("nama5", Nama5);
parameter.put("no_iden5", IdKtp5);
parameter.put("no_telp5", Telp5);
parameter.put("warga5", Warga5);

```

```

parameter.put("jk5", Jk5);

parameter.put("nama6", Nama6);
parameter.put("no_iden6", IdKtp6);
parameter.put("no_telp6", Telp6);
parameter.put("warga6", Warga6);
parameter.put("jk6", Jk6);

parameter.put("nama7", Nama7);
parameter.put("no_iden7", IdKtp7);
parameter.put("no_telp7", Telp7);
parameter.put("warga7", Warga7);
parameter.put("jk7", Jk7);

parameter.put("nama8", Nama8);
parameter.put("no_iden8", IdKtp8);
parameter.put("no_telp8", Telp8);
parameter.put("warga8", Warga8);
parameter.put("jk8", Jk8);

parameter.put("nama9", Nama9);
parameter.put("no_iden9", IdKtp9);
parameter.put("no_telp9", Telp9);
parameter.put("warga9", Warga9);
parameter.put("jk9", Jk9);

parameter.put("nama10", Nama10);
parameter.put("no_iden10", IdKtp10);
parameter.put("no_telp10", Telp10);
parameter.put("warga10", Warga10);
parameter.put("jk10", Jk10);

RequestHandler_2 jParser_2 = new RequestHandler_2();

return jParser_2.sendPostRequest(Server.URL +
"inp_booking.php", parameter);

}

@Override
protected void onPostExecute(String result) {

    super.onPostExecute(result);
    pDialog.dismiss(); }}

```

5.5.15. Halaman Konfirmasi Pembayaran Android

Halaman konfirmasi pembayaran ini berfungsi untuk mengkonfirmasi pembayaran user yang sudah membayar nominal yang sudah ditentukan oleh admin. Berikut ini desain dari halaman konfirmasi pembayaran gunung.

Gambar 5.18. Halaman Konfirmasi Pembayaran Android

5.5.15.1. Coding Program Halaman Konfirmasi Pembayaran Android

Coding ini ada di semua halaman konfirmasi pembayaran. Disini hanya mengambil salah satu coding dari semua halaman konfirmasi pembayaran yang disediakan sistem.

```

package asus.com.gunung.booking_gunung;

import ...

public class Search_Konfirmasi_Bayar extends AppCompatActivity
implements View.OnClickListener {
    private SimpleDateFormat dateFormatter;
    private TextView tvDateResult;
    private Button btDatePicker, button_simpan, btn_bukti_pembayaran;
    private DatePickerDialog datePickerDialog;
    private TextView tvDateResult1;
    private Button btDatePicker1;
    private static final int CAMERA_REQUEST = 1888;
    private ImageView imageView_bukti_pembayaran;
    private static final int MY_CAMERA_PERMISSION_CODE = 100;

    public static final String TAG_NAMAG = "namag";
    public static final String TAG_SUCCESS = "success";
    public static final String TAG_MESSAGE = "message";
    EditText kdb, nml, bank, norek, anm;
    String kdbl, nml1, bank1, norek1, anm1, tgl1, img_base64;
    int success;
    String tag_json_obj = "json_obj_req";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_search_konfirmasi_bayar);
    }
}

```

```

dateFormatter = new SimpleDateFormat("dd-MM-yyyy", Locale.US);

tvDateResult = (TextView) findViewById(R.id.tgl);
btDatePicker = (Button) findViewById(R.id.bt_date);
btDatePicker.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        showDateDialog();
    }
});
//namag = getIntent().getStringExtra(TAG_NAMAG);

kdb = findViewById(R.id.kodebooking);
nml = findViewById(R.id.nominal);
anm = findViewById(R.id.an);
bank = findViewById(R.id.bank);
norek = findViewById(R.id.norek);

imageView_bukti_pembayaran =
findViewById(R.id.image_bukti_pembayaran);

btn_bukti_pembayaran = findViewById(R.id.bukti_pembayaran);
btn_bukti_pembayaran.setOnClickListener(new
View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    public void onClick(View view) {
        if (checkSelfPermission(Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new
String[]{Manifest.permission.CAMERA}, MY_CAMERA_PERMISSION_CODE);
        } else if
(checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, 1);

        } else {
            // Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
            // startActivityForResult(cameraIntent,
CAMERA_REQUEST);
            selectImage(Search_Konfirmasi_Bayar.this);
        }
    }
});

button_simpan = findViewById(R.id.btn_simpan);
button_simpan.setOnClickListener(this);
}

private void selectImage(Context context) {
    final CharSequence[] options = {"Take Photo", "Choose from
Gallery", "Cancel"};

    AlertDialog.Builder builder = new AlertDialog.Builder(context);
    builder.setTitle("Choose your profile picture");

    builder.setItems(options, new DialogInterface.OnClickListener()
{

        @Override
        public void onClick(DialogInterface dialog, int item) {

```

```

        if (options[item].equals("Take Photo")) {
            Intent takePicture = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(takePicture, 0);

        } else if (options[item].equals("Choose from Gallery"))
{
            Intent pickPhoto = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
            startActivityForResult(pickPhoto, 1);//one can be
replaced with any action code

        } else if (options[item].equals("Cancel")) {
            dialog.dismiss();
        }
    }
});
builder.show();
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    // if (requestCode == MY_CAMERA_PERMISSION_CODE) {
    //     if (grantResults[0] == PackageManager.PERMISSION_GRANTED)
    //     {
    //         Toast.makeText(this, "camera permission granted",
Toast.LENGTH_LONG).show();
    //         Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
    //         startActivityForResult(cameraIntent, CAMERA_REQUEST);
    //     } else {
    //         Toast.makeText(this, "camera permission denied",
Toast.LENGTH_LONG).show();
    //     }
    // }
    if (requestCode == 1) {
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {

        } else {
            Toast.makeText(getApplicationContext(), "Permission
Denied", Toast.LENGTH_LONG).show();
        }
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    // if (requestCode == CAMERA_REQUEST && resultCode ==
Activity.RESULT_OK) {
    //
    //     Bitmap photo = (Bitmap) data.getExtras().get("data");
    //     Bitmap resizePhoto = Bitmap.createScaledBitmap(photo,
500, 500, true);
    //     String base64 = encodeImage(photo != null ? photo :
null);
    //     img_base64 = base64;
    //     Log.d("PHOTO", photo != null ? photo.toString() : null);
    //     Log.d("PHOTO_BASE64", photo != null ? base64 : null);
}

```

```

//          imageView_bukti_pembayaran.setImageBitmap(resizePhoto);
//      }

      if (resultCode != RESULT_CANCELED) {
          switch (requestCode) {
              case 0:
                  if (resultCode == RESULT_OK && data != null) {
                      Bitmap selectedImage = (Bitmap)
data.getExtras().get("data");
                      Bitmap resizePhoto =
Bitmap.createScaledBitmap(selectedImage, 500, 500, true);
                      String base64 = encodeImage(selectedImage !=
null ? selectedImage : null);
                      img_base64 = base64;
                      Log.d("CAMERA", resizePhoto != null ? base64 :
null);

imageView_bukti_pembayaran.setImageBitmap(resizePhoto);
                      }

                      break;
              case 1:
                  if (resultCode == RESULT_OK && data != null) {
                      Uri selectedImage = data.getData();
                      String[] filePathColumn =
{MediaStore.Images.Media.DATA};
                      if (selectedImage != null) {
                          Cursor cursor =
getContentResolver().query(selectedImage,
                              filePathColumn, null, null, null);
                          if (cursor != null) {
                              cursor.moveToFirst();

                              int columnIndex =
cursor.getColumnIndex(filePathColumn[0]);
                              String picturePath =
cursor.getString(columnIndex);
                              Bitmap image =
BitmapFactory.decodeFile(picturePath);
                              Bitmap resizeImage =
Bitmap.createScaledBitmap(image, 500, 500, true);
                              String base64 = encodeImage(resizeImage
!= null ? resizeImage : null);
                              img_base64 = base64;
                              Log.d("GALLERY", resizeImage != null ?
base64 : null);

imageView_bukti_pembayaran.setImageBitmap(resizeImage);
                              cursor.close();
                          }
                      }

                      break;
                  }
            }
        }

private String encodeImage(Bitmap bm) {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    bm.compress(Bitmap.CompressFormat.JPEG, 100, baos);
    byte[] b = baos.toByteArray();
    String encImage = Base64.encodeToString(b, Base64.DEFAULT);
}

```

```

        return encImage;
    }

    private void showDateDialog() {
        Calendar newCalendar = Calendar.getInstance();
        datePickerDialog = new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {

            @Override
            public void onDateSet(DatePicker view, int year, int
monthOfYear, int dayOfMonth) {

                Calendar newDate = Calendar.getInstance();
                newDate.set(year, monthOfYear, dayOfMonth);

                tvDateResult.setText(dateFormatter.format(newDate.getTime()));
            }

        }, newCalendar.get(Calendar.YEAR),
newCalendar.get(Calendar.MONTH),
newCalendar.get(Calendar.DAY_OF_MONTH));

        datePickerDialog.show();
    }

    @Override
    public void onClick(View v) {
        new addData().execute();
        Toast.makeText(Search_Konfirmasi_Bayar.this, "Input Berhasil",
Toast.LENGTH_SHORT).show();
        startActivity(new Intent(this, MainMenuActivity.class));
    }

    @SuppressWarnings("StaticFieldLeak")
    class addData extends AsyncTask<Void, Void, String> {

        ProgressDialog pDialog;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            pDialog = new ProgressDialog(Search_Konfirmasi_Bayar.this);
            pDialog.setMessage("Menambahkan Data");
            pDialog.setIndeterminate(true);
            pDialog.setCancelable(true);
            pDialog.show();
        }

        @Override
        protected String doInBackground(Void... params) {
            tgl1 = tvDateResult.getText().toString();
            kdb1 = kdb.getText().toString();
            nml1 = nml.getText().toString();
            anm1 = anm.getText().toString();
            bank1 = bank.getText().toString();
            norek1 = norek.getText().toString();

            HashMap<String, String> parameter = new HashMap<String,
String>();
            parameter.put("kdb", kdb1);
            parameter.put("tgl", tgl1);
            parameter.put("nml", nml1);

```



```

        parameter.put("anm", anm1);
        parameter.put("bank", bank1);
        parameter.put("norek", norek1);
        parameter.put("img_bukti", img_base64);

        RequestHandler_2 jParser_2 = new RequestHandler_2();

        return jParser_2.sendPostRequest(Server.URL +
"inp_konfirmasi.php", parameter);

    }

    @Override
    protected void onPostExecute(String result) {

        super.onPostExecute(result);
        pDialog.dismiss();
    }
}

```

5.5.16. Halaman Input Lokasi Android

Halaman input lokasi ini berfungsi untuk menginputkan koordinat posisi para user sebelum melakukan pendakian. Berikut ini desain dari halaman input lokasi.

Gambar 5.19. Halaman Input Lokasi Android

5.5.16.1. Coding Program Halaman Input Lokasi Android

```

package asus.com.gunung;

import...

public class InputLokasiActivity extends AppCompatActivity
implements View.OnClickListener {

    EditText edtNama, kd_booking;
    TextView latitude, longitude;
    Button btnInput;
}

```

```

public static final String TAG_SUCCESS = "success";
public static final String TAG_MESSAGE = "message";
String tag_json_obj = "json_obj_req";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_input_lokasi);

    edtNama = findViewById(R.id.editTextNamaPendaki);
    kd_booking = findViewById(R.id.editTextKdBooking);
    latitude = findViewById(R.id.textLatitude);
    longitude = findViewById(R.id.textLongitude);
    btnInput = findViewById(R.id.btnInputData);

    btnInput.setOnClickListener(this);

    GPSTracker mGPS = new GPSTracker(this);
    if (mGPS.canGetLocation()) {
        mGPS.getLocation();
        latitude.setText(String.valueOf(mGPS.getLatitude()));
        longitude.setText(String.valueOf(mGPS.getLongitude()));
    } else {
        Toast.makeText(this, "Unable to find",
            Toast.LENGTH_LONG).show();
        System.out.println("Unable");
    }
}

private class MyLocationListener implements LocationListener {

    @Override
    public void onLocationChanged(Location location) {
        if (location != null) {

            latitude.setText(String.valueOf(location.getLatitude()));

            longitude.setText(String.valueOf(location.getLongitude()));
        }
    }

    @Override
    public void onClick(View v) {
        if (v == btnInput) {
            callAsynchronousTask();
        }
    }

    public void callAsynchronousTask() {
        final Handler handler = new Handler();
        Timer timer = new Timer();
        TimerTask doAsynchronousTask = new TimerTask() {
            @Override
            public void run() {
                handler.post(new Runnable() {
                    public void run() {
                        try {
                            String KdBooking =
kd_booking.getText().toString();
                            String Nama =
edtNama.getText().toString();
                            String Latitude =

```

```

latitude.getText().toString();
        String Longitude =
longitude.getText().toString();

        addData(KdBooking, Nama, Latitude,
Longitude);
    } catch (Exception e) {
        // TODO Auto-generated catch block
    }
    });
    }
};
timer.schedule(doAsynchronousTask, 0, 300000); //execute in
every 300000 ms, 1 detik = 1000 ms
}

private void addData(final String kd_booking, final String
nama, final String latitude, final String longitude) {
    StringRequest strReq = new
StringRequest(Request.Method.POST, Server.URL + "inp_lokasi.php",
new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            //JSONObject j = null;
            try {
                JSONObject jsonObj = new JSONObject(response);
                int success = jsonObj.getInt(TAG_SUCCESS);
                if (success == 0) {
                    Toast.makeText(getApplicationContext(),
jsonObj.getString(TAG_MESSAGE), Toast.LENGTH_LONG).show();

                } else {
                    Toast.makeText(InputLokasiActivity.this,
"Input Berhasil", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                //e.printStackTrace();
            }
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    }) {

        @Override
        protected Map<String, String> getParams() {
            // Posting parameters to login url
            Map<String, String> params = new HashMap<String,
String>();
            params.put("kd_booking", kd_booking);
            params.put("nama_pendaki", nama);
            params.put("latitude", latitude);
            params.put("longitude", longitude);

            return params;
        }
    }
};

RequestQueue requestQueue = Volley.newRequestQueue(this);
strReq.setRetryPolicy(new DefaultRetryPolicy(50000,
DefaultRetryPolicy.DEFAULT_MAX_RETRIES,

```

```
DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));
    requestQueue.add(strReq);
    ApplicationController.getInstance().addToRequestQueue(strReq,
tag_json_obj);
    }
}
```